

ATM Networks: Performance Modelling and Evaluation Volume 2

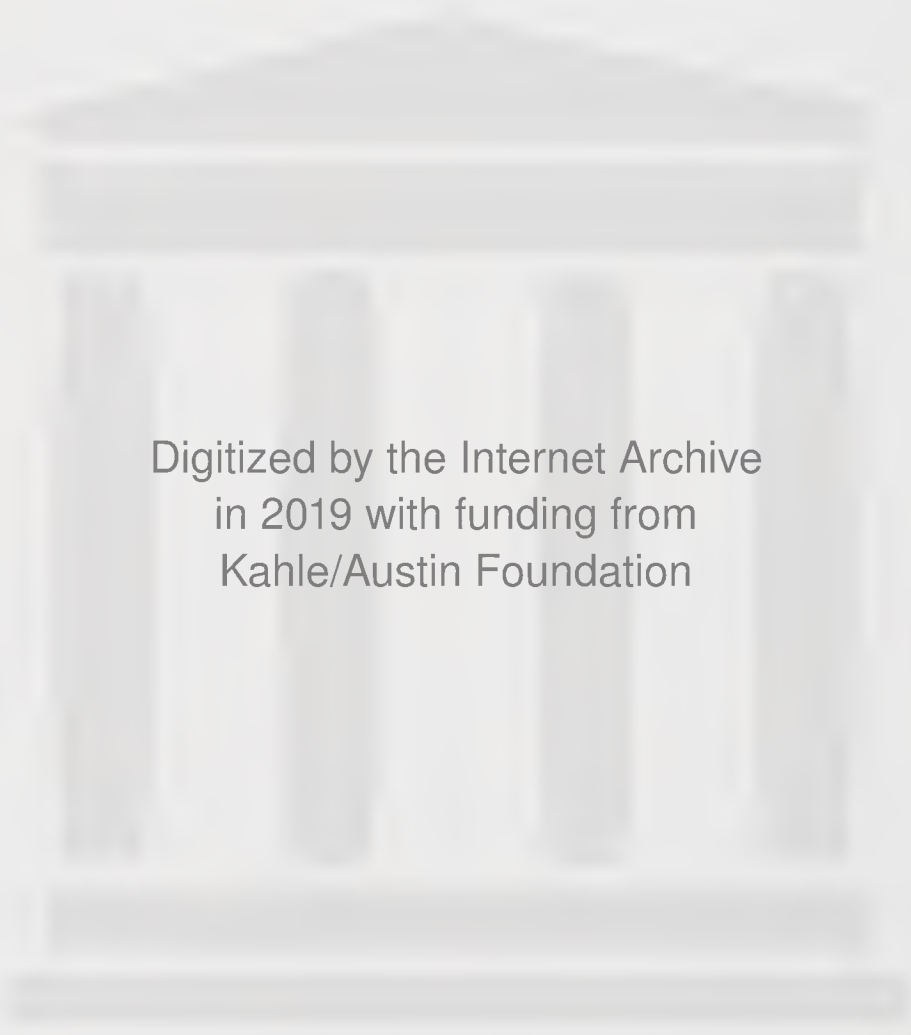
Edited by
Demetres Kouvatsos



IFIP



CHAPMAN & HALL



Digitized by the Internet Archive
in 2019 with funding from
Kahle/Austin Foundation

ATM Networks

IFIP – The International Federation for Information Processing

IFIP was founded in 1960 under the auspices of UNESCO, following the First World Computer Congress held in Paris the previous year. An umbrella organization for societies working in information processing, IFIP's aim is two-fold: to support information processing within its member countries and to encourage technology transfer to developing nations. As its mission statement clearly states,

IFIP's mission is to be the leading, truly international, apolitical organization which encourages and assists in the development, exploitation and application of information technology for the benefit of all people.

IFIP is a non-profitmaking organization, run almost solely by 2500 volunteers. It operates through a number of technical committees, which organize events and publications. IFIP's events range from an international congress to local seminars, but the most important are:

- the IFIP World Computer Congress, held every second year;
- open conferences;
- working conferences.

The flagship event is the IFIP World Computer Congress, at which both invited and contributed papers are presented. Contributed papers are rigorously refereed and the rejection rate is high.

As with the Congress, participation in the open conferences is open to all and papers may be invited or submitted. Again, submitted papers are stringently refereed.

The working conferences are structured differently. They are usually run by a working group and attendance is small and by invitation only. Their purpose is to create an atmosphere conducive to innovation and development. Refereeing is less rigorous and papers are subjected to extensive group discussion.

Publications arising from IFIP events vary. The papers presented at the IFIP World Computer Congress and at open conferences are published as conference proceedings, while the results of the working conferences are often published as collections of selected and edited papers.

Any national society whose primary activity is in information may apply to become a full member of IFIP, although full membership is restricted to one society per country. Full members are entitled to vote at the annual General Assembly, National societies preferring a less committed involvement may apply for associate or corresponding membership. Associate members enjoy the same benefits as full members, but without voting rights. Corresponding members are not represented in IFIP bodies. Affiliated membership is open to non-national societies, and individual and honorary membership schemes are also offered.

ATM Networks

Performance Modelling and Analysis

Volume 2

Edited by

Demetres D. Kouvatsos

Computer System Modelling Research Group

University of Bradford

UK

Published by Chapman & Hall on behalf of the
International Federation for Information Processing (IFIP)



CHAPMAN & HALL

London · Weinheim · New York · Tokyo · Melbourne · Madras

Thomas J. Bata Library
TRENT UNIVERSITY
PETERBOROUGH, ONTARIO

T 15105-25

A 86

1996

V.2

Published by Chapman & Hall, 2-6 Boundary Row, London SE1 8HN, UK

Chapman & Hall, 2-6 Boundary Row, London SE1 8HN, UK
Chapman & Hall GmbH, Pappelallee 3, 69469 Weinheim, Germany
Chapman & Hall USA, 115 Fifth Avenue, New York, NY 10003, USA
Chapman & Hall Japan, ITP-Japan, Kyowa Building, 3F, 2-2-1 Hirakawacho, Chiyoda-ku, Tokyo 102, Japan
Chapman & Hall Australia, 102 Dodds Street, South Melbourne, Victoria 3205, Australia
Chapman & Hall India, R. Seshadri, 32 Second Main Road, CIT East, Madras 600 035, India

First edition 1996

© 1996 IFIP


Printed in Great Britain by St Edmundsbury Press, Bury St Edmunds, Suffolk

ISBN 0 412 79200 1

Apart from any fair dealing for the purposes of research or private study, or criticism or review, as permitted under the UK Copyright Designs and Patents Act, 1988, this publication may not be reproduced, stored, or transmitted, in any form or by any means, without the prior permission in writing of the publishers, or in the case of reprographic reproduction only in accordance with the terms of the licences issued by the Copyright Licensing Agency in the UK, or in accordance with the terms of licences issued by the appropriate Reproduction Rights Organization outside the UK. Enquiries concerning reproduction outside the terms stated here should be sent to the publishers at the London address printed on this page.

The publisher makes no representation, express or implied, with regard to the accuracy of the information contained in this book and cannot accept any legal responsibility or liability for any errors or omissions that may be made.

A catalogue record for this book is available from the British Library

 Printed on permanent acid-free text paper, manufactured in accordance with ANSI/NISO Z39.48-1992 and ANSI/NISO Z39.48-1984 (Permanence of Paper).

To Mihalis and Maria

CONTENTS

Preface	ix
Sponsorship	xi
Participants in the Review Process	xiii
PART ONE Traffic Models and Characterisation	1
1 Validation and tuning of an MPEG-1 video model <i>M. Conti and E. Gregori</i>	3
2 Using maximum entropy principle for output burst characterization of an ATM switch <i>T.S. Rao, S.K. Bose and K.R. Srivathsan</i>	22
3 Using Markovian models to replicate real ATM traffics <i>Å. Arvidsson and C. Lind</i>	39
PART TWO Traffic and Congestion Control	55
4 A congestion control mechanism for connectionless services offered by ATM networks <i>S. Halberstadt, D. Kofman and A. Gravey</i>	57
5 ATM traffic prediction using FIR neural networks <i>Z. Fan and P. Mars</i>	74
6 Analysis, simulation and experimental verification of the throughput of GCRA based UPC functions for CBR streams <i>F. Hoeksema, J. Kroeze and J. Witters</i>	92
7 When is traffic dispersion useful? A study on equivalent capacity <i>E. Gustafsson and G. Karlsson</i>	110
PART THREE Routing and Optimisation	131
8 A comparison of pre-planned routing techniques for virtual path restoration <i>P.A. Veitch, D.G. Smith and I. Hawker</i>	133
9 Virtual path bandwidth control versus dynamic routing control <i>I. Papanikos, M. Logothetis and G. Kokkinakis</i>	153
PART FOUR Adaptation Layer and Protocols	173
10 Some simulation results about TCP connections in ATM networks <i>M.A. Jmone Marsan, A. Bianco, R.L. Cigno and M. Munafò</i>	175

PART FIVE Network Management	195
11 Feedback and pricing in ATM networks <i>L. Murphy and J. Murphy</i>	197
PART SIX Models of ATM Switches	213
✓ 12 Geometrical bounds of an output stream of a queue in a model of a two-stages interconnection network: application to the dimensioning problem <i>L. Truffet</i>	215
✓ 13 A diffusion cell loss estimate for ATM with multiclass bursty traffic <i>E. Gelenbe, X. Mang and Y. Feng</i>	233
✓ 14 An integrated approach to evaluating the loss performance of ATM switches <i>S. Montagna, R. Paglino and J.F. Meyer</i>	249
15 Strictly nonblocking operation of 3-stage Clos switching networks <i>F.K. Liotopoulos and S. Chalasani</i>	269
16 Performance analysis of buffered Banyan ATM switch architectures <i>D. Kouvatsos, J. Wilkinson, P. Harrison and M. Bhabuta</i>	287
PART SEVEN Bandwidth and Admission Control	325
17 Markov chain animation technique applied to ATM bandwidth derivation and tandem switches <i>J.M. Griffiths and J.M. Pitts</i>	327
18 A scheme for multiplexing ATM sources <i>J. Naudts, G. De Laet and X.W. Yin</i>	342
✓ 19 Shaping of video traffic to optimise QoS and network performance <i>A. Dagiuklas, M. Ghanbari and B.J. Tye</i>	358
PART EIGHT Performance Modelling Studies	379
20 Study of the performance of an ATM CLOS switching network based on the composite technique <i>G. Fiche, Cl. Le Palud and S. Rouillard</i>	381
21 A study of the fairness of the fast reservation protocol <i>L. Cerdà, J. García and O. Casals</i>	400
✓ 22 Efficient simulation of consecutive cell loss in ATM networks <i>V.F. Nicola and G.A. Hagesteijn</i>	414
23 On the accelerated simulation of VBR virtual channel multiplexing in a single-server first-in-first-out buffer <i>M.J. Tunnicliffe and D.J. Parish</i>	431
Index of contributors	441
Keyword index	443

Preface

Asynchronous Transfer Mode (ATM) networks are widely considered to be the new generation of high speed communication systems both for broadband public information highways and for local and wide area private networks. Over recent years there has been a great deal of progress in research and development of ATM technology, but there are still many interesting and important problems to be resolved such as traffic characterisation and control, routing and optimisation, ATM switching techniques and provision of specified quality of service.

This book presents twenty-three research papers, both from industry and academia, reflecting latest original contributions in the theory and practice of performance modelling and analysis of ATM networks worldwide. These papers were selected, subject to peer review, from those submitted as expanded and revised versions out of eighty-nine shorter papers presented at the Third IFIP Workshop on "Performance Modelling and Evaluation of ATM Networks", July 2-6, 1995, Craiglands Hotel, Ilkley, West Yorkshire, UK. At least three referees drawn from the Scientific Committee and externally were involved in the evaluation process of each paper.

The research papers were classified into seven parts covering the following topics: Traffic Models and Characterisation, Traffic and Congestion Control, Routing and Optimisation, Adaptation Layer and Protocols, Network Management, Models of ATM Switches, Bandwidth and Admission Control and Performance Modelling Studies.

Part One on "Traffic Models and Characterisation" includes three papers and is concerned with modelling and performance implications of multiplexed streams of bursty and correlated traffic in ATM networks. New analytic traffic models are proposed, focusing, respectively, on the characterisation of ATM traffic generated by Variable Bit Rate (VBR) video applications and the determination of output burst length of an ATM switch via entropy maximisation. Moreover, a validation study is presented relating to Markovian models replicating real ATM traffic flows.

Part Two on "Traffic and Congestion Control" addresses fundamental objectives such as guaranteed network performance, traffic prediction and management and contracted quality of service. This part brings together four papers describing analytic and simulation studies on ATM traffic and congestion control mechanisms. The works are based on flow control at connectionless layer combined with dynamic bandwidth allocation, Finite Impulse Response (FIR) neural networks, User Parameter Control (UPC) functions and the strategy of traffic dispersion.

Part Three on “Routing and Optimisation” focuses on the inherent problems of many services envisaged for ATM networks involving information transfer from one to one or one to many recipients for multimedia applications. It includes two papers which devise appropriate performance metrics and carry out rigorous comparisons involving pre-planned routing techniques for virtual path restoration as well as control schemes on virtual path bandwidth and dynamic routing under both static and dynamic traffic conditions.

Part Four on “Adaptation Layer and Protocols” reports a single study discussing detailed simulation experiments on the adaptability and performance issues of the Transport Control Protocol (TCP) when running over high speed ATM networks. **Part Five** on “ATM Management” presents one paper concerning with the provision of traffic loss guarantees in economically efficient ATM networks by means of an iterative pricing algorithm incorporating, as a dynamic feedback signal, a load dependent price per usage unit of network resources.

Part Six on “Models of ATM Switches” consists of five papers which describe analytic methodologies and cost-effective algorithms for the performance evaluation of various ATM switch architectures such as Multistage Interconnection Networks (MINs), shared output buffer queues and 3-stage clos switching networks. The methodologies are based on discrete-time Markovian analysis, diffusion approximation approach, maximum entropy principle and traffic flow formalism for non-blocking operations. Such robust and reliable tools and techniques are of great value towards the derivation of new closed-form expressions and bounds for typical performance measures such as queue length distributions, cell-loss (and blocking) probabilities and end-to-end delays.

Part Seven on “Bandwidth and Admission Control” is concerned with novel methodologies for ATM bandwidth and performance optimisation, call connection control and traffic shaping. This part includes three papers which apply numerical simulations and also analytical techniques using theoretical arguments and an iterative Markov chain scheme.

Finally, **Part Eight** on “Performance Modelling Studies” includes four papers dealing with various ATM performance modelling and evaluation issues. The first two papers apply, respectively, analytical methods relating to a composite technique for an ATM clos switching network and Markov Chain solutions for fast reservation protocols. The last two papers deal with the important topic of accelerated simulation techniques for ATM networks.

I would like to end this forward by expressing my thanks to IFIP TC6 and Working Groups WG 6.3 and WG 6.4 for sponsoring the 3rd Workshop on the Performance Modelling and Evaluation of ATM Networks and to British Computer Society Performance Engineering Specialist Group, Performance Engineering Section of BT Labs., UK, Telematics International Ltd., UK, Departments of Computing, of Electrical Engineering and of Mathematics, University of Bradford, Engineering and Physical Sciences Research Council (EPSRC), UK, for their support. My thanks are also extended to the members of the Scientific Committee and external referees for their invaluable and timely reviews.

Demetres Kouvatsos

Sponsorship

IFIP TC6

IFIP WG 6.3 on the Performance of Communication Networks

IFIP WG6.4 on Communication Networks

Also supported by

The British Computer Society
Performance Engineering Specialist Group

The Performance Engineering Section of BT Labs., UK

Telematics International Ltd., UK

The Department of Computing
University of Bradford

The Department of Electrical Engineering
University of Bradford

The Department of Mathematics
University of Bradford

The Engineering and Physical Sciences Research Council (EPSRC), UK

Chair

Demetres Kouvatsos, Bradford, UK

Scientific Committee

John Arnold, GPT Ltd., U.K.
Åke Arvidsson, Karlskrona/Ronneby, Sweden
Simonetta Balsamo, Pisa, Italy
Monique Becker, Evry, France
Chris Blondia, Antwerpen, Netherlands
Herwig Bruneel, Ghent, Belgium
Olga Casals, Catalonia, Spain
Marco Conti, CNUCE, Italy
Laurie Cuthbert, London, U.K.
Nico van Dijk, Amsterdam, Netherlands
Lorenzo Donatiello, Bologna, Italy
Erol Gelenbe, Duke, U.S.A.
Richard Gibbens, Cambridge, U.K.
Peter Harrison, London, U.K.
Boudewijn Haverkort, Twente, Netherlands
Christoph Herrmann, Philips, Germany
Ilias Iliadis, IBM Zürich, Switzerland
László Jereb, Budapest, Hungary
Peter Key, BT, UK
Ulf Körner, Lund, Sweden
Paul Kühn, Stuttgart, Germany
Isi Mitrani, Newcastle, U.K.
Nicholas Mitrou, Athens, Greece
Arne Nilsson, North Carolina, U.S.A.
Raif Onvural, IBM, U.S.A.
Achille Pattavina, Milano, Italy
Harry Perros, North Carolina, U.S.A.
Michal Pióro, Warsaw, Poland
Ramón Puigjaner, Illas Balears, Spain
Guy Pujolle, Paris, France
Douglas Reeves, North Carolina, U.S.A.
John Silvester, Southern California, U.S.A.
Andreas Skliros, Telematics Int. Ltd., U.K.
Geoff Smith, Strathclyde, U.K.
Otto Spaniol, Aachen, Germany
Ioannis Stavrakakis, Boston, U.S.A.
Yutaka Takahashi, Kyoto, Japan
Don Towsley, Massachusetts, USA
Phuoc Tran-Gia, Würzburg, Germany
Yannis Viniotis, North Carolina, U.S.A.
Mike Woodward, Loughborough, U.K.
Hideaki Yamashita, Tokyo, Japan

Participants in the Review Process

Marco Ajmone-Marsan
Manuel Alvarez-Campana
John S Arnold
Årk Arvidsson
Simonetta Balsamo
Monique Becker
Steven Blake
Chris Blondia
Laura J. Bottomley
Bruno Bouton
Herwig Bruneel
Brian Bunday
Nigel Burton
Olga Casals
Marco Conti
Tadeusz Czachórski
Neil Davies
Lorenzo Donatiello
Nick Duffield
Zbigniew Dziong
Jürgen Enssle
Yanke Fan
Georges Fiche
Jean-Michel Fourneau
Maurice Gagnaire
Erol Gelenbe
Nicolas D. Georganas
Panos Georgatsos
Mohammad Ghanbari
John M. Griffiths
Peter G. Harrison
Boudewijn R. Haverkort
Christoph Herrmann
Fokke W. Hoeksema
Ilias Iliadis
László Jereb
Mourad Kara
Gunnar Karlsson
Roger Kaye
Frank Kelly
Peter Key
Leila Kloul
Daniel Kofman
George E. Konstantoulakis
Demetres Kouvatsos
Koenraad Laevens
János Levendovsky
Qin Li
Fotios K. Liotopoulos
Renato Lo Cigno

Michael Logothetis
Xiaowen Mang
María Jesús Manso Godino
Brian G. Marchent
Phil Mars
Carl McCrosky
Elena Medova
John Mellor
Nikos M. Mitrou
Sergio Montagna
John Murphy
Jan Naudts
James Ni
Arne Nilsson
Raif Onvural
Tolga Örs
David J. Parish
Achille Pattavina
Nihal Pekergin
Ferhan Pekergin
Harry G. Perros
Christopher Phillips
Michal Pióro
Jonathan M. Pitts
Rob Pooley
Francesco Potorti
Ramón Puigjaner
Guy Pujolle
Douglas S. Reeves
I. E. G. Richardson
Martyn J. Riley
Miguel Rios
Michael Ritter
John Schormans
Geoff Smith
Ioannis Stavrakakis
Bart Steyaert
Zhili Sun
Don Towsley
Phuoc Tran-Gia
Laurent Truffet
Boris Tsybakov
Paul A. Veitch
Yannis Viniotis
Dietmar Wagner
Johan Witters
Hideaki Yamashita
Hirofumi Yokoi
Yury Zlotnikov

PART ONE

Traffic Models and Characterisation

Validation and Tuning of an MPEG-1 Video Model

Marco Conti, Enrico Gregori

CNUCE, Institute of National Research Council

Via S. Maria 36 - 56126 Pisa - Italy, Phone: +39-50-593111

Fax: +39-50-589354, E-mail: {M.Conti, E.Gregori}@cnuce.cnr.it

Abstract

Variable Bit Rate (VBR) video traffic is expected to become one of the major traffic sources for high-speed networks. Although the modeling of VBR video sources has recently received significant attention, there is currently no widely accepted model which lends itself to mathematical analysis. This paper addresses the problem of characterizing the traffic generated by VBR video applications. Specifically, we define an analytically tractable model for the traffic generated by an MPEG-1 encoder. An extensive validation of this model is carried out by analyzing its suitability to capture the statistical behavior of a wide variety of MPEG-1 sources ranging from movies, sports events, talk shows, etc. We show that our model, with an adequate tuning of its parameters, is able to provide an accurate representation of these different kinds of MPEG-1 sources.

Keywords

Variable bit rate video, MPEG, ATM, statistical multiplexing, Markov chain,

1 INTRODUCTION

Recent technological advances in fiber optics and switching systems have provided the technological basis for the development of high-capacity Broadband-Integrated Services Digital Networks (*B-ISDNs*), which are capable of supporting transmission speeds of several hundred Mbps [1]. This enormous potential for fast and massive information transport should be

Work carried out in the framework of CNR coordinated projects "Gestione del traffico VBR in ambiente interconnesso", and "Sistemi distribuiti real-time per il supporto di applicazioni multimediali".

able to support not only the traditional data and voice services, but also a variety of new applications, including the transport of images, teleconferencing, moving video, and large volumes of interactive computer data. Asynchronous Transfer Mode (*ATM*) is the transfer technique for the implementation of such B-ISDNs, due to its efficiency and flexibility [1].

Although significant research effort has focused on the development of efficient information multiplexing schemes for the diversified B-ISDN/ATM environment, most of the practical problems related to real-time applications remain unsolved and not addressed by the ATM Forum [1, 2].

Variable Bit Rate (VBR) video is currently by far the most interesting and challenging real-time application. A VBR encoder attempts to keep the quality of video output constant and at the same time reduces bandwidth requirements since only a minimum amount of information has to be transferred. On the other hand, as VBR video traffic is both highly variable and delay sensitive, high-speed networks (e.g., ATM) are generally implemented by assigning peak rate bandwidths to VBR video applications, and by using the residual bandwidth for non-real-time traffic. This approach may however be inefficient. To define bandwidth allocation schemes which provide an adequate QoS for VBR applications and minimize the wastage of bandwidth, the effects of the video applications on the network must be investigated.

VBR video generates a traffic with complex characteristics which cannot be effectively described in terms of traditional traffic models. Developing accurate and analytically tractable representation schemes for real-time traffic will provide a basis for the development of efficient multiplexing schemes and increased utilization of networking resources.

While the modeling of VBR video sources has recently received significant attention [1, 2, 4, 10, 11, 12, 13, 14], there is currently no widely accepted model which lends itself to mathematical analysis. Furthermore, new video compression standards, such as the *MPEG* family [7, 8, 15], are emerging. In this paper we focus on the MPEG-1 coding algorithm. The MPEG-1 standard specifies the coding algorithm for full motion video information with an output peak rate in the 1.5-2 Mb/s range. Starting from the analysis of the trace of the movie "Star Wars" encoded with MPEG-1 algorithm we propose an analytically tractable model [6]. In this paper we validate this model by analyzing its suitability to capture the statistical behavior of a wide variety of MPEG-1 sources ranging from movies, sports events, talk shows, etc. The output of an MPEG-1 coder, depending on the class of traffic, exhibits a very different statistical behavior. Movies have, in general, a "heavy tailed" autocorrelation function, while, in TV sports and in "talk shows", the correlation disappears after a few seconds. We show that the "heavy tailed" behavior has a significant impact on the buffer-size statistics in ATM multiplexers.

Our model, with an adequate tuning of its parameters, is able to provide an accurate representation of these different kinds of MPEG-1 sources. Specifically, by exploiting experimental results, we have identified a fitting procedure which provides a relationship between the behavior of a real source (mainly the tail of its autocorrelation function) and the model-parameter values which provides the best fitting of the source behavior.

The paper is organized as follows: Section 2 presents the characteristics of MPEG-1 sources relevant for our investigation. An MPEG-1 analytical model is described in Section 3 and validated in Section 4. The tuning of the model is investigated in Section 5.

2 MPEG-1 VIDEO SOURCE

An uncompressed video source may generate bits at rates as high as hundreds of Mbps (Mega-bits per second); a few such sources could, in other words, occupy the entire network capacity that is available today. Data compression techniques are therefore employed to reduce the video source bit rate which is transmitted over the network. The resulting traffic is highly variable and dependent on the encoding scheme adopted and on the activity of the movie.

VBR video is currently considered to be the dominant bandwidth-demanding real-time application for high-speed networks in the immediate future. Developing accurate and analytically tractable models for this kind of traffic will thus provide a basis for the design and development of these networks. Before this can be done we need to fully understand the characteristics of the video source.

2.1 MPEG-1 coding scheme

MPEG-1 is a specification for coding moving pictures, developed by the ISO Joint Motion Pictures Experts group. The standard is well suited for a large range of video applications at a variety of bit rates. A combination of video and audio information, particularly for “movie” applications, can also be compressed. Typical compression ratios are in the range of 50:1 to 200:1 [3]. The algorithm is *asymmetrical*; that is, it requires more computational complexity to compress video than to decompress it. Applications well suited for this are those that require the frequent use of decompression, but for which compression is only done once. A very good example of this is Video On Demand (*VOD*).

MPEG-1 is an interframe coder. Coders in this class exploit, in addition to intraframe coding, the temporal redundancy between adjacent frames by predicting the next frame from the current one. A key feature that distinguishes MPEG-1 from previous coding algorithms is *bidirectional temporal prediction*. For this type of prediction, some of the frames are encoded using two reference frames, one in the past and one in the future. This results in higher compression gains.

As indicated above, when applying MPEG-1 to video, one of three different coding modes can be used for each frame. The terminology used for the resulting frame is related to the mode used and is as follows:

- *I-frame*: intra frame coded, i.e. coded with JPEG.
- *P-frame*: predictive coded with reference to a past picture.
- *B-frame*: bidirectional predictive coded.

I-frames provide access points for random access but only with moderate compression. Predictive coded frames are also generally used as a reference for future B-frames. Type B frames provide the highest amount of compression but require both a past and future reference prediction. In addition, B-frames are never used as reference frames.

In the encoded sequence, as shown in Figure 1, the frames are arranged into *groups*. In this case a group consists of 12 frames - one I-frame, three P-frames and eight B-frames. Figure 1 also shows the relationship between the frames. We can see that I-frames are independent, P-frames are predicted, and B-frames are bidirectionally predicted.

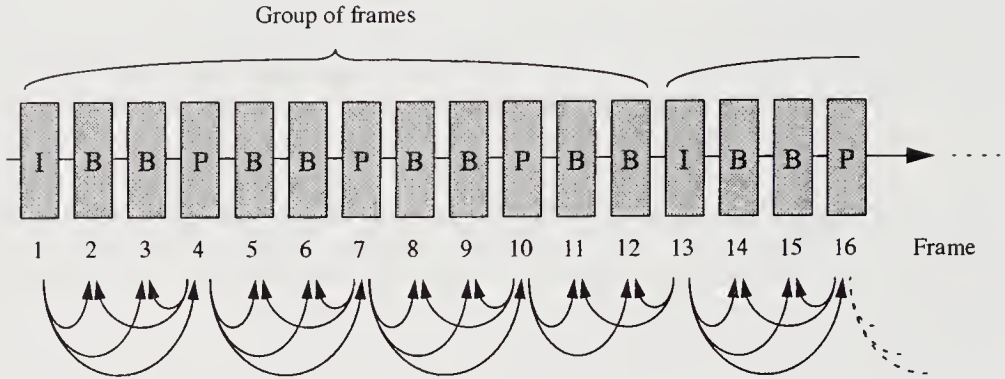


Figure 1 A sequence of MPEG-1 frames and their relationship.

2.2 Statistics of MPEG-1 coded movies

Figure 2 shows a small extraction from the output of the MPEG-1 coded movie *Star Wars* released by M. Garret at Bellcore. Specifically, frames are coded into groups of twelve frames as defined in Figure 1 (i.e., the frame pattern is IBBPBBPBBPBB).

As shown in Figure 2, the bandwidth required to transmit consecutive frames is highly variable and very much depends on the frame types, I, P and B. Furthermore, as expected (due to the coding scheme algorithm) the shape of the output is repeated every twelve frames.

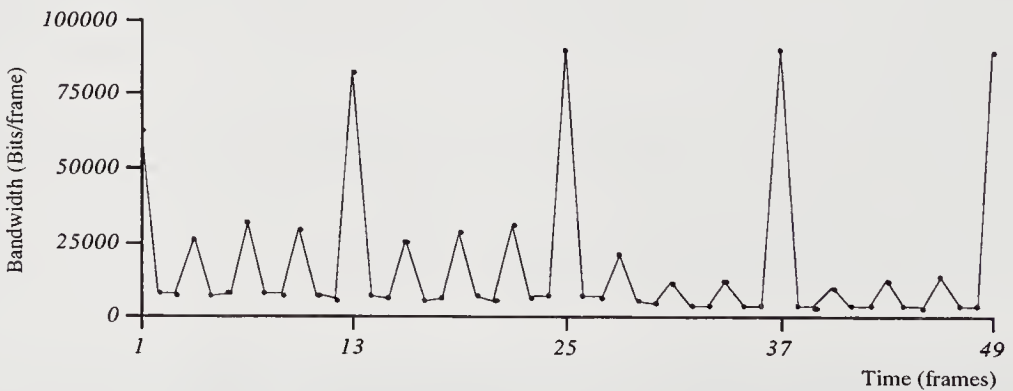


Figure 2 Part of the MPEG-1 coder trace, revealing group length and frame pattern.

Statistical analysis indicates that the output of an MPEG-1 encoder should be described by three partially correlated¹ submodels where each submodel describes the output process corresponding to one frame-type. This leads to a model with a very large state space.

The model space complexity is reduced by avoiding a separate representation for the var-

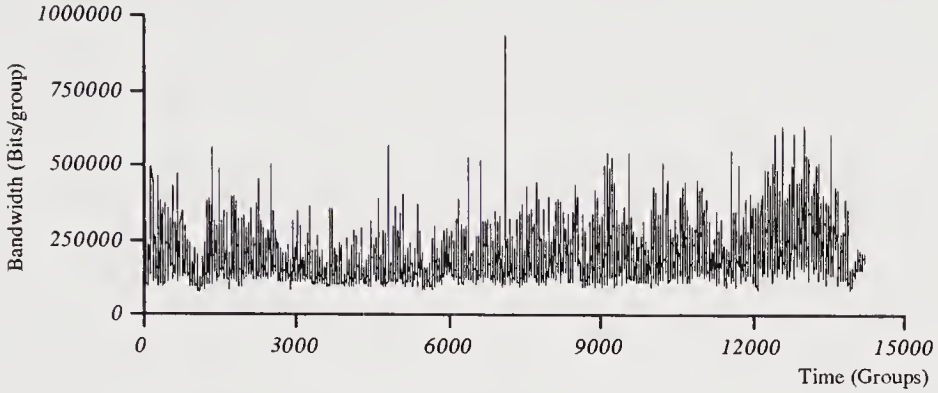


Figure 3 The aggregate sequence.

ious frame-types. Specifically, this is obtained by considering a different time scale, in which the time unit is the group (i.e., a sequence IBBPBBPBBPBB) and the bit rate per time unit is the sum of the amount of bandwidth generated by all the frames in a group. One group is in this case equal to 12 frames and each frame is generated every $1/24$ th second. The resulting sequence is hereafter named *aggregate sequence*.

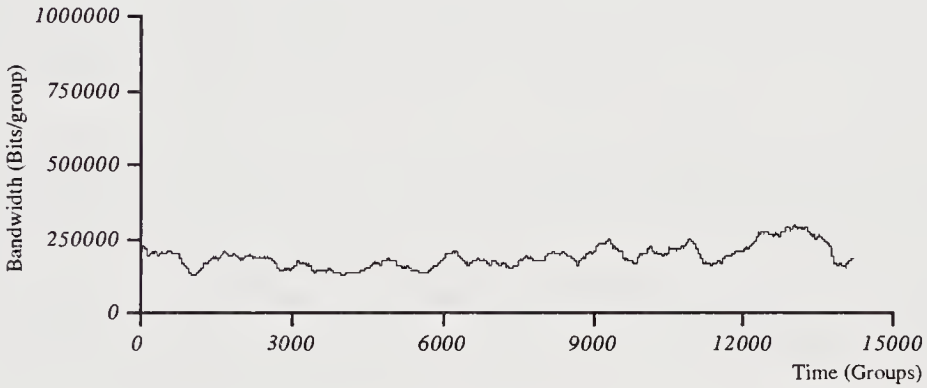


Figure 4 Low frequency component of the aggregate sequence.

Figure 3 shows a plot of the *aggregate sequence* generated by an MPEG-1 coder with the Star Wars as a source. A time unit, on the x-axis, is equal to 0.5 seconds, i.e., a group inter-arrival.

The bit rate of consecutive frames shows that the bandwidth changes in a rapid but bursty way. However there is also a slowly changing underlying structure. This *low frequency* underlying structure of the sequence can be better highlighted by passing the aggregate sequence through a moving average filter of length W . The result of this filtering process

1. A precise analysis of the correlation between the three different subsequences is presented in [6].

with a window size $W=300$ groups (i.e. two and a half minutes), is shown in Figure 4. These characteristics of the aggregate sequence can be highlighted better by observing its autocorrelation function.

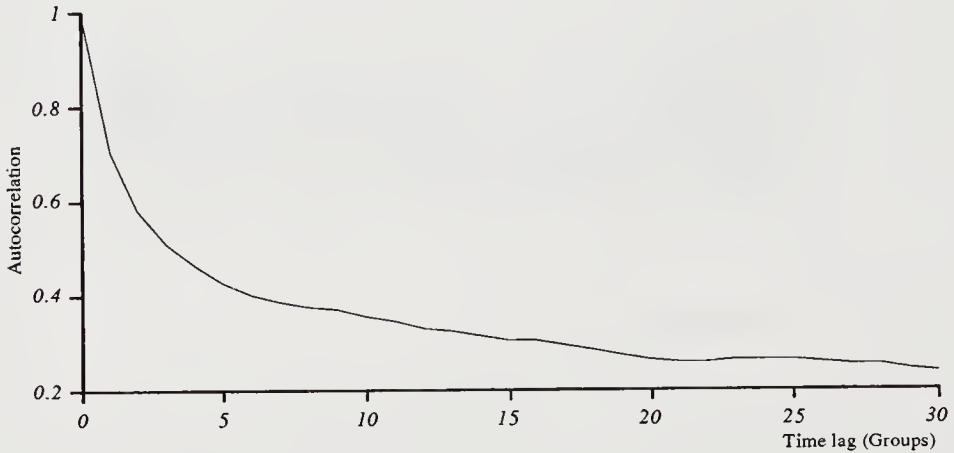


Figure 5 Short range dependencies

The autocorrelation function for the aggregate sequence is plotted in Figures 5 and 6, showing the short ($0 \leq n \leq 30$) and long range ($n > 30$) dependencies, respectively. Figure 5 shows the existence of a strong short-range dependence for time lags below approximately 30 groups, which corresponds to 15 seconds. In this range the autocorrelation function drops quickly (the autocorrelation with $n = 30$ is about 0.2). However, after this sharp initial decrease, as shown in Figure 6, it takes a very long time before the autocorrelation function drops to zero. Specifically, Figure 6 highlights the existence of a significant long-range dependence which lasts for time lags up to 3500 groups i.e., about 29 minutes. The tail of the autocorrelation function decreases slowly. Similar behavior of the autocorrelation function have been observed in [15] for other sequences.

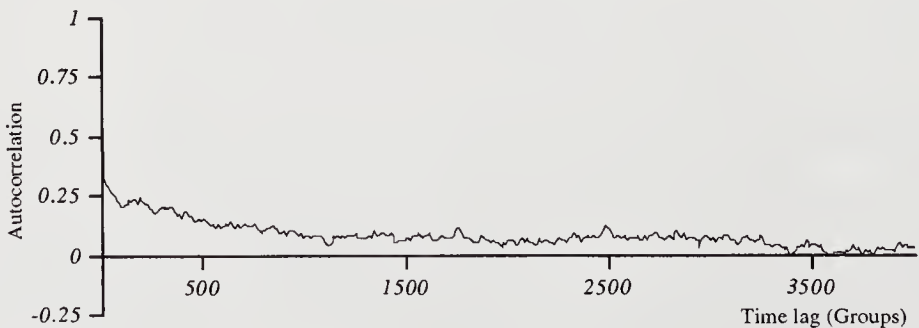


Figure 6 Long range dependencies

3 MPEG-1 MODELING

Figures 5 and 6 show that in the aggregate sequence there are both short-range dependencies which last for around 20-30 groups (some seconds), and long-range dependencies which last for thousands of groups (some minutes). In order to capture both types of dependencies a bidimensional Markov chain $\{L_k, H_k, k \geq 0\}$ is used, in which $\{L_k | k \geq 0\}$ is used to represent the long term correlation, while $\{H_k | k \geq 0\}$ represents the short term correlation. Specifically, in our model the process $\{H_k | k \geq 0\}$ describes the bit rate per group of an MPEG-1 encoder. To avoid unnecessary complexity (in the state space $\{H_k | k \geq 0\}$) we quantize the bit rate information into a number of levels. The number of quantization levels for the process $\{H_k | k \geq 0\}$ will hereafter be denoted by N (i.e., $H_k \in \{0, 1, 2, \dots, N-1\}$). The question of which quantization method should be used is not discussed here. For us it seemed natural to use uniform quantization. For this reason, let max and min denote the maximum and minimum bit rates observed in the aggregate sequence. The possible bit rates between max and min are quantized with a constant step size $\Delta = (max - min) / N$, resulting in the actual bit rate of the source equal to $j \cdot \Delta + min$ where j is the quantization level holding the property $0 \leq j \leq N-1$.

To represent the low-frequency component of an MPEG source, a modulating process $\{L_k | k \geq 0\}$ is included in the model as well ($L_k \in \{0, 1, 2, \dots, M-1\}$). The transitions in the Markov chain occur every time unit (i.e., a group interarrival), while the process $\{L_k\}$ changes its state on average only after 70-100 time units.

The process we want to model now takes the form $\{L_k, H_k, k \geq 0\}$, where $L_k \in \{0, \dots, M-1\}$ is the status of the low frequency process corresponding to the k th group, and $H_k \in \{0, \dots, N-1\}$ is the corresponding state in the high frequency process.

The transition probabilities of the Markov chain, denoted by $p_{ij,lm}$

$$p_{ij,lm} = P(L_k = l, H_k = m | L_{k-1} = i, H_{k-1} = j). \quad (3.1)$$

are estimated from an MPEG-1 trace through the procedure reported below. To explain the procedure better we use $\{f_0, f_1, f_2, \dots\}$ to indicate the frame sequence in the original sequence.

Procedure for the computation of transition probabilities $p_{ij,lm}$

1. Produce the aggregate sequence (*high frequency sequence*) $\{a_0, a_1, a_2, \dots\}$, where a_i denotes the bit rate of the i -th group:

$$a_i = \sum_{j=0}^{11} f_{12i+j}. \quad (3.2)$$

The number 12 refers to the group length used by the MPEG-1 coder (see Figure 1).

2. Produce the aggregate filtered sequence (*low frequency sequence*) $\{\bar{a}_0, \bar{a}_1, \bar{a}_2, \dots\}$, where \bar{a}_i denotes the bit rate of the i -th group in the filtered sequence. \bar{a}_i is obtained by passing the aggregate sequence through a moving average filter of length W :

$$\bar{a}_i = \frac{1}{W} \sum_{j=-\lceil W/2 \rceil}^{\lceil W/2 \rceil-1} a_{i+\lceil W/2 \rceil+j}. \quad (3.3)$$

3. Quantize the high and low frequency sequences into M and N uniform levels respectively. Number the levels from 0 to $M - 1$ and from 0 to $N - 1$.
4. Using *low* and *high frequency sequences*, measure the 1-step transition probability matrix \mathbf{P} consisting of

$$p_{ij,lm} = p^{(1)}_{ij,lm} = P(L_k = l, H_k = m | L_{k-1} = i, H_{k-1} = j),$$

where L_k is the k th element in the quantized low-frequency sequence, H_k is the k th element in the quantized high-frequency sequence, $i, l \in \{0, \dots, M-1\}$ and $j, m \in \{0, \dots, N-1\}$.

The models presented throughout are obtained with parameters $M=8$ and $N=8$. In [6] we showed that these parameter values represent a good compromise between precision and complexity.

Specifically, by applying the fitting procedure to the Star Wars sequence, our Markov chain has the transition matrix \mathbf{P} shown in (3.4). Submatrices \mathbf{A}_{ii} of \mathbf{P} represent the probabilities that the process does not change the low frequency level in a transition i.e., $P\{H_k = j, L_k = i | H_{k-1} = l, L_{k-1} = i\}$. Submatrices $\mathbf{A}_{i+1,i}$ ($\mathbf{A}_{i-1,i}$) represent the probabilities that the process moves to the next (previous) low frequency level in a transition $P\{H_k = j, L_k = i | H_{k-1} = l, L_{k-1} = i+1\}$ ($P\{H_k = j, L_k = i | H_{k-1} = l, L_{k-1} = i-1\}$).

$$\mathbf{P} = [p_{ij,lm}] = [\mathbf{A}_{il}]$$

$$= \begin{bmatrix} \mathbf{A}_{00} & \mathbf{A}_{01} & & & & & & & \\ & \mathbf{A}_{10} & \mathbf{A}_{11} & \mathbf{A}_{12} & & & & & \\ & & \mathbf{A}_{21} & \mathbf{A}_{22} & \mathbf{A}_{23} & & & & 0 \\ & & & \mathbf{A}_{32} & \mathbf{A}_{33} & \mathbf{A}_{34} & & & \\ & & & & \mathbf{A}_{43} & \mathbf{A}_{44} & \mathbf{A}_{45} & & \\ & & & & & \mathbf{A}_{54} & \mathbf{A}_{55} & \mathbf{A}_{56} & \\ & & & & & & \mathbf{A}_{65} & \mathbf{A}_{66} & \mathbf{A}_{67} \\ & & & & & & & \mathbf{A}_{76} & \mathbf{A}_{77} \end{bmatrix} \quad (3.4)$$

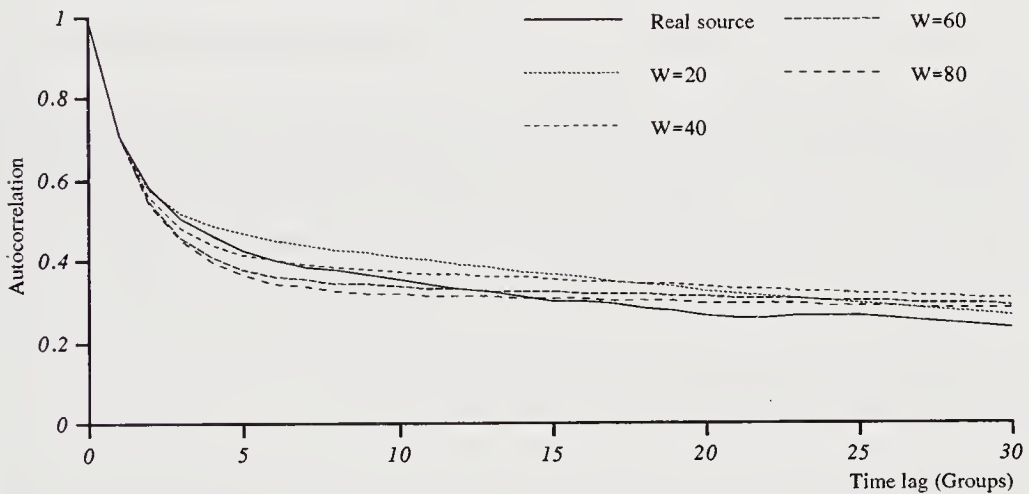
4 MODEL VALIDATION

In this section we analyse the characteristics of the model to see whether it can imitate the behavior of "Star Wars". Table 1 outlines the basic statistics of the MPEG-1 Star Wars trace, where, "original" and "aggregate" indicate the sequences with the frame and the group as time units (see Section 2.2), respectively. Figure 7 plots the real source autocorrelation function ($r(n)$) and for four different models constructed with various moving average window lengths, W . Time lags used for the calculation range from 0 to 30 groups. The plot thus com-

Table 1 Basic statistics for the MPEG-1 “Star Wars” movie.

Measure	Original Sequence	Aggregate Sequence
Mean bandwidth, μ	15598 bits/frame	187185 bits/group
Standard deviation, σ	18165 bits/frame	72468 bits/group
Coefficient of variation, μ/σ	1.16	0.39
Peak bandwidth	185267 bits/frame	932710 bits/group
Minimum bandwidth	476 bits/frame	77754 bits/group
Peak/mean bandwidth	11.88	4.98

compares the short-range dependence of the real source, and different parametrizations of the model. The model constructed with $W = 20$, has a stronger short-range dependence than the real source. It has, however, a faster decay. Even if $r(n)$ of the model is still above the

**Figure 7** Comparison of the real source’s and the model’s short-range dependencies ($M = 8$, $N = 8$).

real source one for a time lag equal to 30 groups, the difference is smaller than for $n = 10$. As the value of W is increased, the autocorrelation function of the model tends to fall off in the beginning but it decreases more slowly. The model with $W = 40$ is a good example to emphasize this behavior. For n less than 7, the short-range dependence of the model takes on values lower than the real source. For time lags beyond this point, the plot shows that $r(n)$ of the model decays more slowly than for the real source. The autocorrelation functions for models constructed with $W = 60$ and $W = 80$ follow the same pattern.

We know that the long-range dependence of MPEG-1 coded VBR video is very strong. Figure 8 compares the autocorrelation function of the model and the real source for time lags of 0 up to 2000 groups. Several values of W have been used. A model constructed with $W = 100$ has a long-range dependence which is not as strong as that of the real source for n

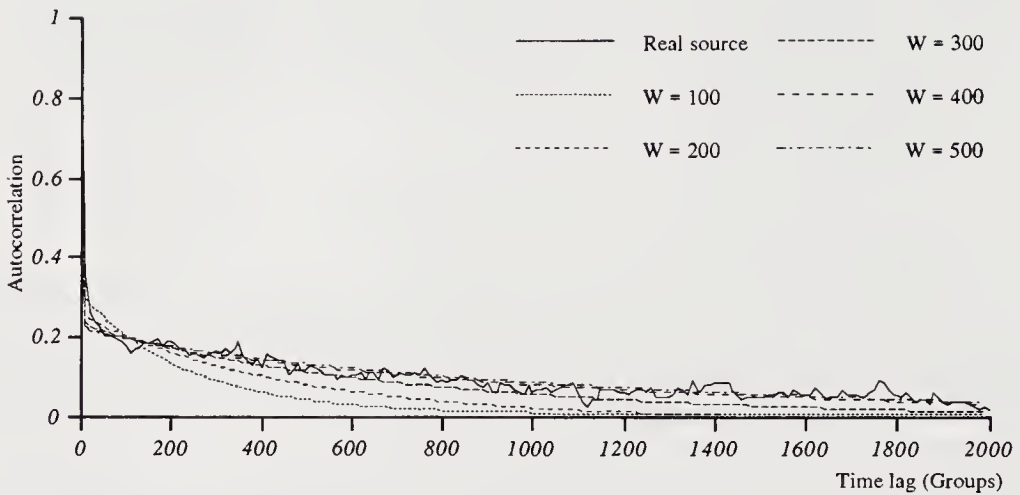


Figure 8 Comparison of the real source's and the model's long-range dependence.

greater than 100. It also reaches zero at a time lag approximately equal to 1100, which is much earlier than in the real source. $r(n)$ for the other models plotted in Figure 8 tells us that the long-range dependence of the model tends to get stronger, and thus approaches the real source, as W is increased. For example, a model constructed with $W = 300$ matches the long-range dependence of the real source better than if it is created with $W = 200$. At the same time we know, from the previous subsection, that a higher value of W implies a weaker short-range dependence.

4.1 Importance of the Long-Term Dependencies

In the previous section we have shown that our model, depending on the setting of the W parameter, is able to precisely capture either short- or long-range dependencies. In this section we investigate the importance of both type of dependencies in the study of the statistical multiplexing. Specifically, the aim of this analysis is to study the smoothing in the traffic profile obtained by the superposition of several VBR video sources depending on the type of correlations existing in traffic.

Multiplexing of VBR video sources is complex, as these applications have low tolerance towards network congestion. Although sufficient buffer capacity may be available, excessive buffering may not be possible, due to the resulting unacceptable delays. In this section we therefore investigate the queueing time distribution experienced by VBR video traffic as a function of the bandwidth reserved for each source. As shown before (see Table 1), the peak rate for our MPEG source corresponds to a bandwidth level equal to $c=7$, while the average is about bandwidth level equal to 0.53. Below we investigate the delay experienced by VBR video traffic by assuming that the bandwidth allocated for each source is about twice the average, i.e., $c=1$. The results reported in Figure 9 were obtained by studying via simulation the queueing delay distribution in a single server queueing system with a deterministic service time, FIFO, and input traffic generated by s independent and identically dis-

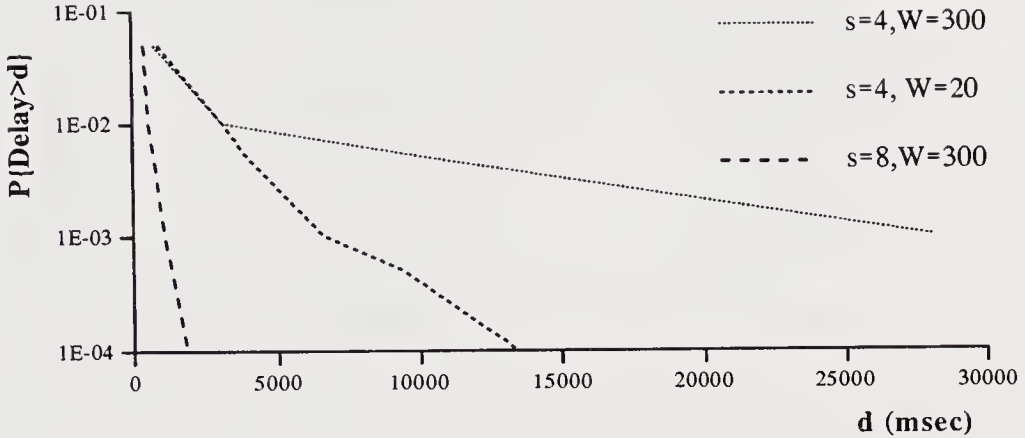


Figure 9 Tail of delay distribution, $c=1.0$

tributed MPEG-1 sources. Note that we use our analytical model to study the statistical multiplexing problem as from a single real trace is impossible to obtain reliable statistical estimates. Furthermore, identically distributed sources can not be obtained by using different traces.² On the other hand our model can provide the number of i.i.d traces required to obtained statistics with the desired precision.

Figure 9 shows that a 53% network utilization and acceptable delays can be achieved if at least eight sources are multiplexed. In addition, the figure clearly indicates that the tail estimated with $W=20$ is extremely underestimated in the region $(1E-04, 1E-02)$. These results show that the long term correlation affects significantly the tail of the delay distribution also for lightly loaded network (e.g. network utilization in the order of 50%). Partially neglecting it ($W=20$) induces optimistic estimates with errors in the order of 100%! These observations have been confirmed by the extensive analysis presented in [6].

4.2 Model Validation by exploiting other MPEG-1 sources

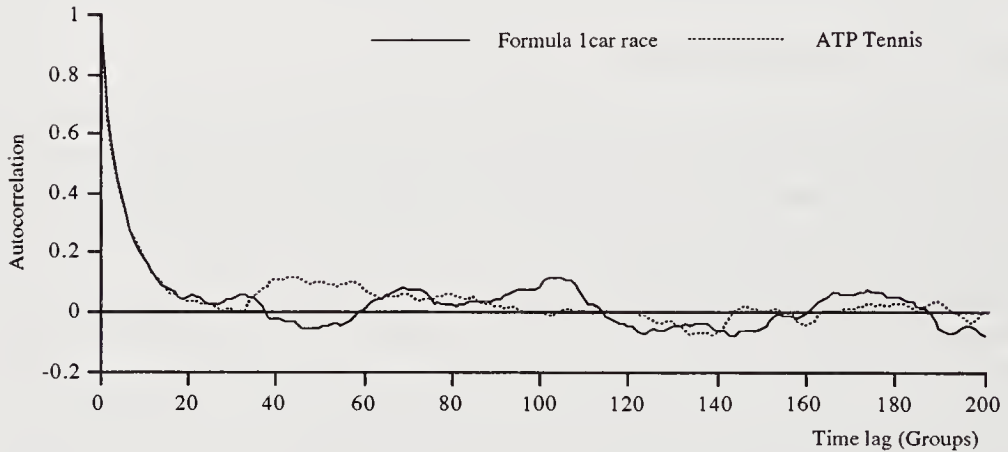
In this section we extend the validation process by analyzing the suitability of the model for capturing the statistical behavior of other MPEG-1 sources. Specifically, we consider a wide variety of sources ranging from movies, sports events, talk shows, etc. The traces related to these sources, encoded with MPEG-1 algorithm with the parameters reported in Table 2, were released by O. Rose [10].

We measured the autocorrelation function of several real traces, and we realized that it is highly variable and depends on the kind of video sequence. We have identified two extreme cases: *movies* and *sports events*. The differences between the two classes are highlighted by Figures 10 and 11. Specifically, the sports events have an autocorrelation function that drops to zero in a few seconds (see Figure 10), and then oscillates around zero.

2. The distribution of the number of bits per groups highly depends on the movie.

Table 2 : Encoder parameters

Encoder input	384 x 288 pel
Colour format	YUV (4:1:1, resolution of 8 bits)
Quantization values	I=10, P=14, B=18
Pattern	IBBPBBPBBPBB
GOP size	12
Motion vector search	'Logarithmic' / 'Simple'
Reference frame	'Original'
Slices	1
Vector / range	Half pel / 10

**Figure 10** Autocorrelation function for sports events.

On the other hand, movies have a *heavy tailed* autocorrelation function. As shown in Figure 11, in “Terminator II”, the correlation between frames disappears after about 40 seconds (80 groups) while up to 4 minutes (500 groups) are necessary to lose the correlation in “Jurassic Park”. Note that the precision on the estimates of the autocorrelation is affected by the relatively small size of the samples (30 minutes).³ This limited amount of data is responsible for the oscillating behavior of the tail of the autocorrelation function.

We now investigate the model’s flexibility and effectiveness in capturing the behavior of the various real sources. For this reason we plot, in a graph for each trace, the autocorrela-

3. It is worth recalling that the autocorrelation function of the “Star Wars” movie drops to zero very slowly, in about 30 minutes. Note that the trace of this movie was related to two hours of video (the whole movie) and thus the estimated tail has very small fluctuations, whereas the new available traces last approximately half an hour and fluctuations in the order of 0.1 make it almost impossible to analyze the tail of the autocorrelation below this value.

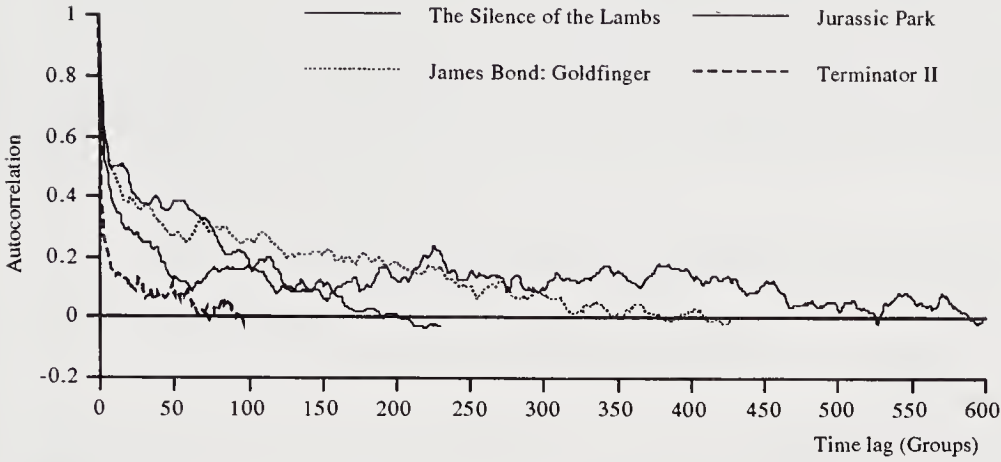


Figure 11 Autocorrelation function for Movies.

tion function of a real source together with the autocorrelation obtained from the model tuned with different window size values, W . As shown before, short- or long-range dependencies can be emphasized by varying the parameter W . The analysis on statistical multiplexing presented in the previous section showed that the long-term correlation significantly affects the tail of the delay distribution. Partially neglecting it leads to optimistic estimates with errors in the order of 100%. For this reason, below we primarily focus on capturing the tail of the long-term correlation.

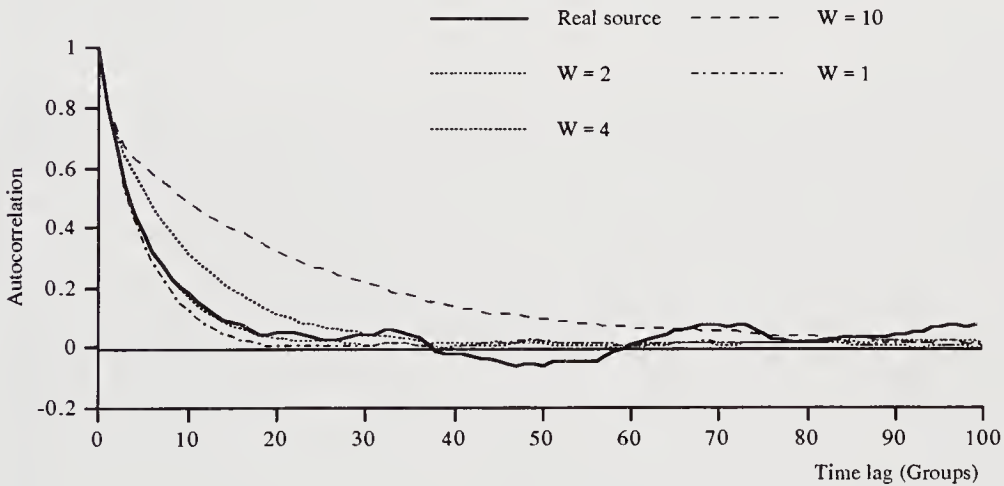


Figure 12 Comparison of the real source's and the model's autocorrelation function for a "Formula-1 car race".

Dependencies between frames in sports events last for a few seconds and are well captured by a filtering with a very small W . Figure 12 clearly indicates that the $W=2$ seems to be the best solution for the “Formula-1 car race” trace.

Figure 11 shows example of movies (i.e., “The Silence of the Lambs” and “James Bond: Goldfinger”) for which the tail of the autocorrelation functions, quickly go down to zero (i.e., the slope of the tail is high). In these cases the low frequency components are captured very well by the model using small window sizes. For example, as shown in Figure 13,

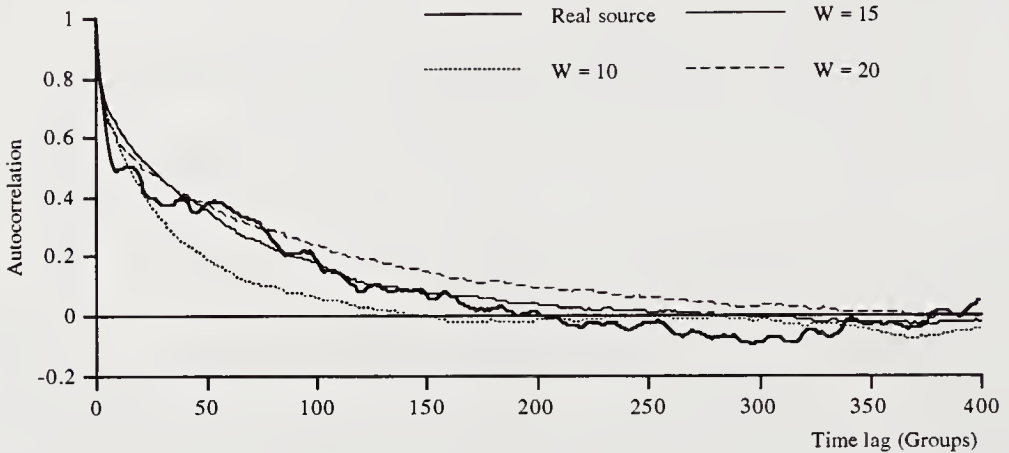


Figure 13 Comparison of the real source’s and the model’s autocorrelation function for the “The Silence of the Lambs”.

$W=15$ provides the best fitting for “The Silence of the Lambs”. A similar behavior can also be observed for the “James Bond: Goldfinger” trace for which we identified $W=60$ as the best window size.

“Terminator II” and “Jurassic Park”, on the contrary, have a slower decrease in the tail of the autocorrelation functions.⁴ We consider that “Terminator II” has a slow autocorrelation-function decay because, although it goes down to zero in about 40 seconds, it begins the second part of the decay starting from a value of autocorrelation equal to 0.15, so that the slope of the tail of the autocorrelation function (that we are interested in) is very low.

As expected, big window sizes are needed to capture the low-frequency component of “Terminator II” and “Jurassic Park”. Specifically, as shown in Figure 14, the model with $W=340$ fits well the tail of the autocorrelation function of the “Jurassic Park” trace, except in the middle where it has a slightly lower value of autocorrelation. Similarly, $W=300$ is identified as the best window size for “Terminator II”.

4. In these cases, however, the fluctuations in the tail of the autocorrelation (computed by a “limited” amount of real data) makes it difficult to find the best window sizes.

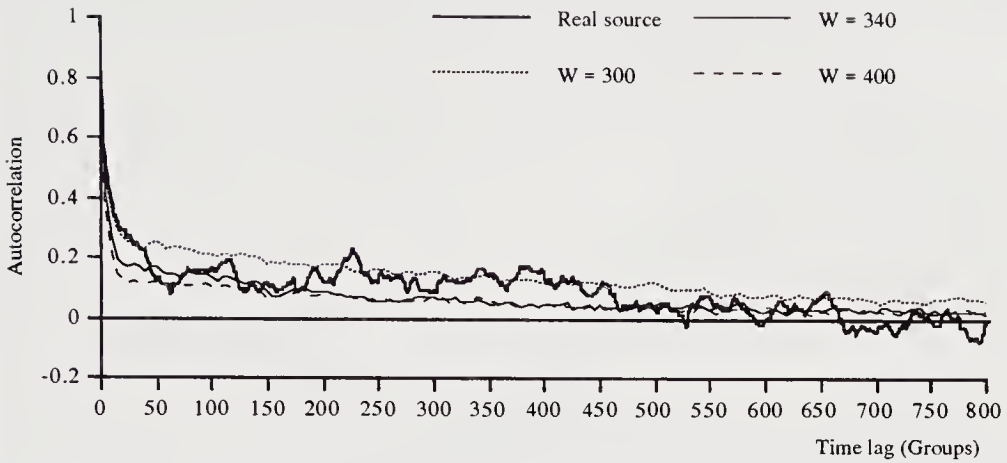


Figure 14 Comparison of the Real source's and the model's autocorrelation function for the movie "Jurassic Park".

5 MODEL TUNING: CHOICE OF THE W PARAMETER VALUE

The analysis presented in the previous section shows that

1. Depending on the type of source (e.g., sports events, movies) the autocorrelation function of the sequence completely changes: the low-frequency component is always significant in movies while it is almost negligible in the sports events.
2. Sources of the same type may have significant differences. For example, in the movies considered in this work, frames become almost independent after 40 seconds for "The Silence of the Lambs", while positive correlations still exist after 10-20 minutes in "Star Wars".

We can thus conclude that neither a general model exists for MPEG-1 sources nor a single model can be defined to characterize (at least) one type of MPEG-1 sources (e.g., movies). The target for MPEG-1 modeling is therefore to define a set of rules to identify, for each MPEG-1 source, the best choice of model-parameter values to capture (as much as possible) the behavior of the source. We have identified this set of rules in the steps 1.- 4 on page 10 of the fitting procedure presented in Section 3. However, the model parameter W very much depends on the source. Identifying the relationship between a real source and the best window size W to capture the tail of its autocorrelation function is still an open issue. In the analysis presented in this work the relationship between the real source and the best W value seems to depend on the slope of the autocorrelation function. Roughly speaking, the autocorrelation function presents two behaviors: a fast decrease in the first frames (e.g., 20-30 frames) and a slower decrease in its tail. A first-order estimate of this behavior can be obtained by fitting each of these portions with a straight line and then approximating the speed in the decrease of the autocorrelation function with the slope, m , of its fitting (straight) line.

As we are mainly interested in capturing long-term correlations, below we apply this

approach to estimate for different movies the decay speed in the tail of their autocorrelation functions. Figures 15, 16 and 17 show the fitting line and the tail of the autocorrelation functions.

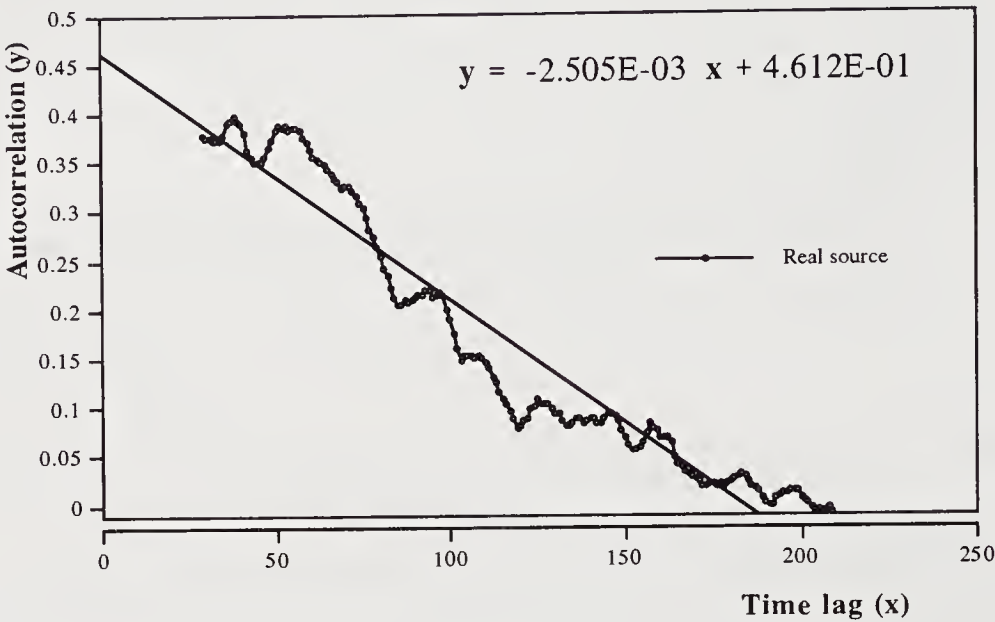


Figure 15 Slope of the “The Silence of the Lambs” autocorrelation.

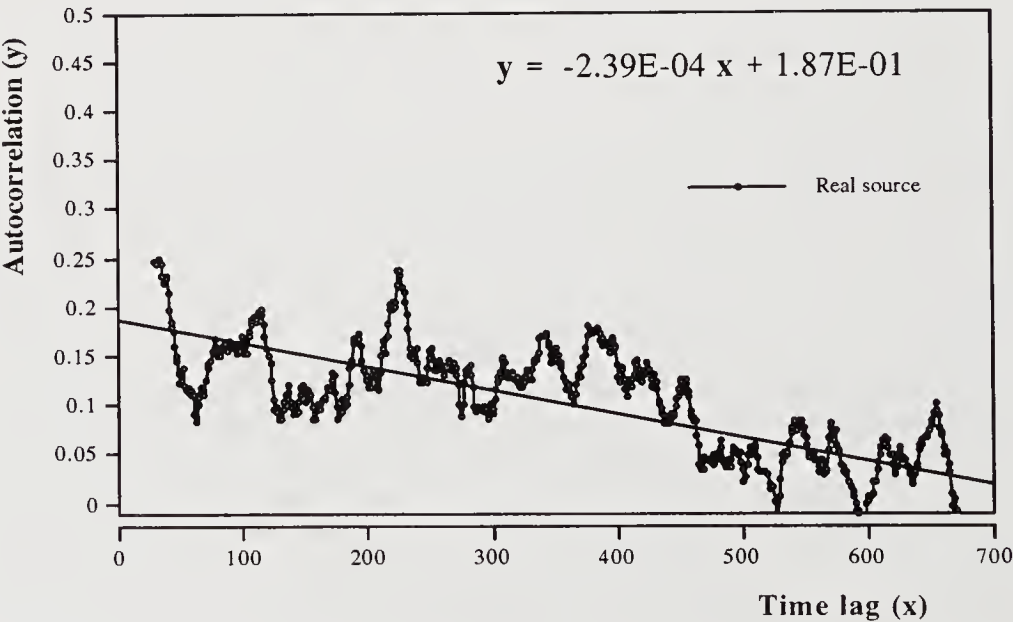


Figure 16 Slope of the “Jurassic Park” autocorrelation.

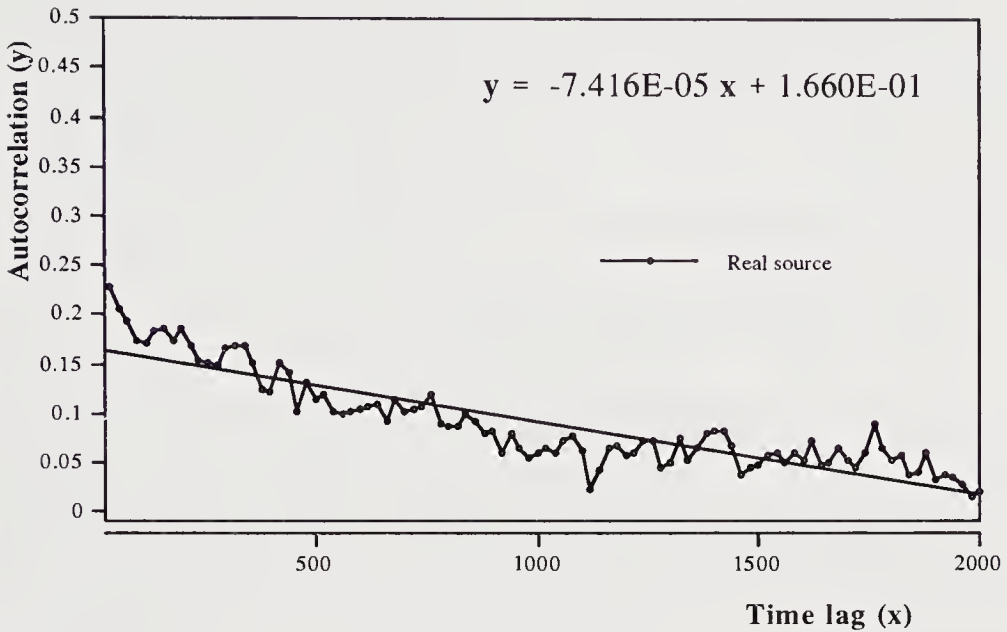


Figure 17 Slope of the “Star Wars” autocorrelation.

function for three different movies (“The Silence of the Lambs”, “Jurassic Park”, “Star Wars”) which exhibit short-, medium- and long-term correlations, respectively. In addition each graph reports the equation of the fitting line.

If we consider the various type of movies, we note that as the slope increases the best window size decreases. For example, for “The Silence of the Lambs” m is in the order of $-2.5 \times 10E-03$ and $W = 15$, for “Jurassic Park” $m \approx -2.4 \times 10E-04$ and $W = 340$, and finally, for “Star Wars” $m \approx -7.4 \times 10E-05$ and $W = 400$. These results indicate that as the slope increases the best window size value decreases, but we still need to provide a mathematical formulation for this relationship.

By plotting the pair of values (m, W) for the different movies, and by fitting these points with a hyperbolic function (see Figure 18) we have identified a heuristic rule: $m \times W \approx \text{constant}$. Hence, we can use the function shown in Figure 18 to identify the best window size for a given source.

6 SUMMARY AND CONCLUSIONS

Modeling VBR video is a difficult task due to the complex statistical characteristics of this type of traffic. In this paper we have considered the modeling of an MPEG-1 source.

We have presented a Markov model which with an adequate tuning of its parameters (mainly the window size), is able to provide an accurate representation of different kinds of MPEG-1 sources. Specifically, the sources considered in this work can be, at least, subdivided into at least two groups: movies and sports events. Dependencies between the frames of movies disappear after minutes, while dependencies in sports events only last for seconds.

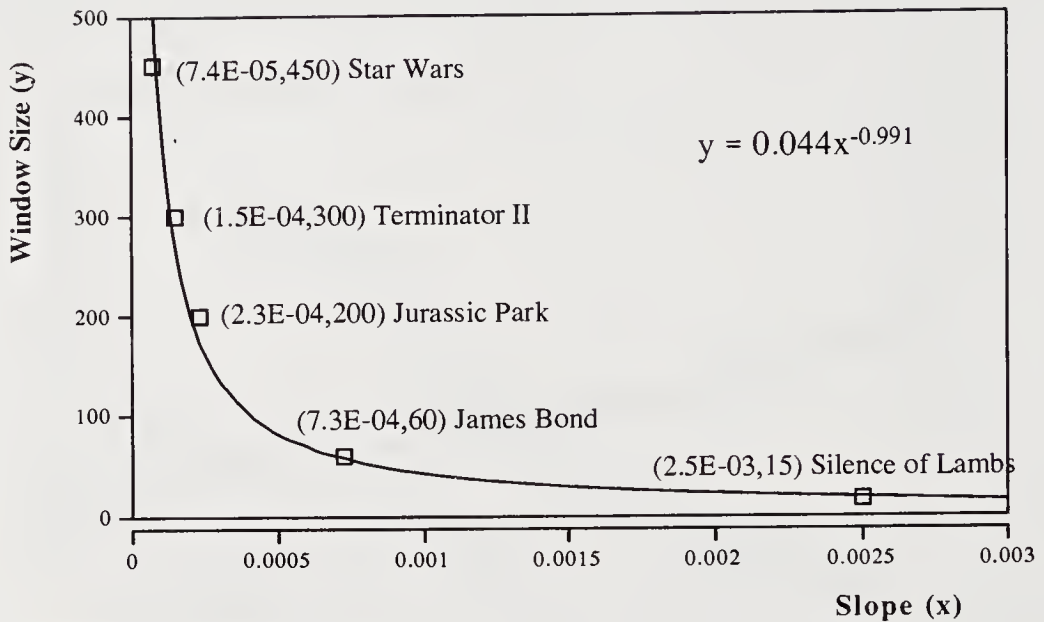


Figure 18 Relationship between the slope of the autocorrelation function, m , and the window size W

Sharp differences also exists among movies (see Figure 4).

We have shown that, at least for statistical multiplexing studies, the tail of the autocorrelation function (i.e., long-term correlations) cannot be neglected. Thus it is impossible to produce a unique characterization of MPEG-1 sources. In fact, depending on the type of source (e.g., sports events, movies) the autocorrelation function of the sequence completely changes. In addition, sources of the same type may have significant differences.

We have presented and validated an approach to produce a precise model of a given MPEG-1 source. The main problem in applying our model is the selection of the window size; the various behaviours of the sources make it impossible to find a single window size for all the cases. Thus we have identified a heuristic rule to overcome the window-size selection problem. Our heuristic is based on the observation that the product of the best window-size value and the decay speed of the autocorrelation function is almost independent of the movie.

Acknowledgements

The authors wish to express their gratitude to the M. Garret (Bellcore) and O. Rose (Wurzburg University) for releasing the MPEG-1 traces.

7 REFERENCES

- [1] R. O. Onvural, *Asynchronous Transfer Mode Networks: Performance Issues*, Artech House, Inc., Norwood, MA, 1994.

- [2] H. Saito, *Teletraffic Technologies in ATM Networks*, Artech House, Inc., Norwood, MA, 1994.
- [3] D. Minoli and R. Keinath, *Distributed Multimedia Through Broadband Communications Services*, Artech House, Inc., Norwood, MA, 1994.
- [4] N. Ohta, *Paket Video: Modeling and signal processing*, Artech House, Inc., Norwood, MA, 1994.
- [5] G. K. Wallace, The JPEG Still Picture Compression Standard, *Communications of the ACM*, Vol. 34, No. 4, April 1991, pp. 30-44.
- [6] M. Conti, E. Gregori, A. Larsson, "Analysis and Modeling of an MPEG-1 Video Source", CNUCE Report, December 1994.
- [7] D. Le Gall, MPEG: A Video Compression Standard for Multimedia Applications, *Communications of the ACM*, Vol. 34, No. 4, April 1991, pp. 46-58.
- [8] L. Chiariglione, "The development of an integrated audiovisual coding standard: MPEG", *IEEE Proceeding*, Vol. 83, No. 2, February 1995, pp. 151-157.
- [9] M. Nomura, T. Fujii and N. Ohta, "Basic Characteristics of Variable Rate Video Coding in ATM Environment", *IEEE Journal on Selected Areas in Communications*, Vol. 7, No. 5, June 1989, pp. 752-760.
- [10] O. Rose, "Statistical Properties of MPEG Video Traffic and Their Impact on Traffic Modeling in ATM Systems", University of Wurzburg, Research Report No.101, February 1995.
- [11] P. Skelly, M. Schwartz and S. Dixit, "A Histogram-Based Model for Video Traffic Behavior in an ATM Multiplexer", *IEEE/ACM Transactions on Networking*, Vol. 1, No. 4, Aug. 1993, pp. 446-459.
- [12] M. R. Frater, P. Tan and J. F. Arnold, "Variable Bit Rate Video Traffic on the Broadband ISDN: Modeling and Verification", *ITC 14*, 1994, pp. 1351-1360.
- [13] M. W. Garret, *Contributions Toward Real-Time Services on Packet Switched Networks*, Ph.D. Dissertation CU/CTR/TR 340-93-20, Columbia University, New York, N.Y., May 1993.
- [14] M. W. Garret, W. Willinger., "Modeling and Generation of Self-Similar VBR Video Traffic", *SIGCOMM'94*, London, Sept. 1994, pp.269-280.
- [15] P. Pancha, M. El Zarki, "MPEG coding for variable bit rate video transmissions" *IEEE Communications Magazine*, May 1994, pp.54-66.
- [16] C. H. Sauer, E. A. MacNair and J. F. Kurose, *The Research Queueing Package Version 2: CMS Users Guide*, IBM Research Report RA-139, Yorktown Heights, N.Y., April 1982.

8 BIOGRAPHY

Marco Conti received the Laurea degree in Computer Science from the University of Pisa, Pisa, Italy, in 1987. In 1987 he joined the Networks and Distributed Systems department of CNUCE, an institute of CNR (the Italian National Research Council). He has worked on modeling and performance evaluation of Metropolitan Area Network MAC protocols. His current research interest include ATM, wireless networks, design, modeling and performance evaluation of computer communication systems.

Enrico Gregori was born in Todi, Italy, in 1955. He received his Laurea degree in Electronic Engineering in April, 1980. Since 1981 he has been working at CNUCE, an institute of CNR (the Italian National Research Council) in the Computer Networks and Distributed Systems department. He has worked in several projects on network architectures and protocols. In 1986 he went on sabbatical at the IBM Research Center in Zurich. His current research interests include the design, modeling and performance evaluation of high speed networks.

Using Maximum Entropy Principle for Output Burst Characterization of an ATM Switch

T. Srinivasa Roa, Sanjay K. Bose, K.R. Srivathsan

*E-Mail: tsr, skb, krsr@iitk.ernet.in Dept. of Elec. Engg.I.I.T. Kanpur - 208 01 INDIA
ph.: +91 512 250697*

Abstract

Maximum Entropy Principle is used in deriving an approximate expression for the burst length of a tagged call at the output of an ATM switch. The statistical multiplexer is approximated as a variable server, infinite buffer queuing system with only cells from the tagged call as clients where each incoming cell sees the server in randomly variable vacations. Numerical experiments are carried out and compared with the simulation results.

Keywords

Statistical Multiplexer, ON-OFF source, Instantaneous bandwidth available, Maximum Entropy principle

1 INTRODUCTION

Asynchronous Transfer Mode (ATM) is expected to be the carrier mode for Broadband Integrated Services Data Networks (B-ISDN). In B-ISDN, different calls will have different call characteristics, like peak rate, average rate, etc. Also different calls will have different QOS requirements, like packet loss, packet delay, etc. The optical fibre communication, perceived to be a suitable media for B-ISDN applications, provides Bit Error Rate (BER) as low as 10^{-9} - 10^{-10} . Hence ATM which provides cell-based connection oriented network service, is an ideal transport for B-ISDN services on low error fibre optic media. Connection-oriented network service is preferred over connection-less network service because the former demands less processing overhead at intermediate switches than the latter.

Due to the "bursty" nature of B-ISDN applications, statistical multiplexing of calls is preferred for its effective utilization of bandwidth and buffer resources. Statistical multiplexing, however causes degradation of QOS parameters like average and standard deviation of cell delay and cell loss due to congestion at intermediate ATM switches. Reactive controls, like end-to-end flow control, are commonly used in low speed networks like X.25. In the ATM environment, reactive congestion controls may not be effective because of the large Bandwidth-distance product. Preventive congestion controls like Call Admission Controls (CAC) and User Parameter Control (UPC) are proposed for avoiding congestion in ATM networks. With Call Admission Controls in place, each intermediate switch in the pre-determined path of the call, is required to determine whether the incoming call can be served with the demanded QOS parameters without effecting the QOS of existing calls. If the call can be accepted, the switch forwards the "call request" to the next switch; otherwise the switch sends "call reject" back to the source, in which case the source may hunt for another route for the call.

Most of the literature in performance modeling and evaluation of ATM networks deal with a single link or an isolated switching node. The end-to-end performance analyses of large-scale Broadband Integrated networks

is essential not only for implementation of Call Admission Control procedures but also for understanding the efficiency and financial viability of the network as a whole. As in any interconnected network, the output of the upstream node will be the input for the next node and hence knowledge of the output characteristics of ATM switch is essential. The exact characterization of the output process of an ATM switch is complex and intractable due to statistical multiplexing of various classes of multimedia traffic. Moreover, some intermediate nodes may be fed by output streams of more than one upstream node.

Most approaches for characterizing the output processes, proposed in the literature are approximations and have only limited applicability in call admission controls and end-to-end performance analyses. Y. Ohba, et al [ohba 91] consider an ATM switch in the presence of three kinds of traffic, GI-stream, Batch arrivals and a set of IPP sources. A transient expression for the queue length distribution at the arrival instants of cells from the GI-stream, is developed. Using that queue length distribution, the waiting time distribution and inter-departure time distributions of cells from a GI-stream are obtained. Even though, in principle, the same transient expression can be used iteratively for obtaining steady state queue length distribution, it may not be practical for larger systems. I. Stavrakakis [stav 91] developed models for bursty traffic when they undergo splitting and merging. Specifically, three different models were proposed and compared for bursty traffic when it is splitted and cells routed into the tagged direction with probability p , and diverted away from the tagged direction with probability $(1-p)$. The merging of bursty traffic is characterised as another bursty process in terms of the probabilities of the queue being empty and the queue being not empty. This also analyzes the output processes at intermediate switches in a system of inter-connected switches, with the following assumptions – the input at any switch is only a fraction, p of the output from a previous switch. i.e. the cell will be sent to the targeted direction with probability p . In case of bursty traffic, the above assumption may not be valid.

In certain B-ISDN applications, jitter is one of the QOS parameters. Specifically for real-time applications like audio, jitter is required to be low, so that proper replay of audio is possible at the destination. W. Matragi, et al [mat 94-I, mat 94-II] modeled the jitter of a call at the output as the difference in queue lengths at the departure instants of consecutive cells. They considered the jitter process for a GI-stream of customers in the presence of a batch arrival process. In [mat 94-I], the Z-transform of the jitter of a GI-stream at the output of a single node is obtained. This is extended in [mat 94-II], for the estimation of end-to-end jitter incurred by a periodic traffic in an ATM network. In [rob 92, boy 92], the influence of jitter on peak rate enforcement and user parameter control algorithms is studied. Due to intermittent clumping of cells, user parameter control algorithms need to be more complex. I. Cidon, et al [cid 94], obtained analytical expressions for messages, maximum cell delay in a message and the number of cells in a message whose delay exceeded pre-specified time thresholds. The analytical expressions obtained here can be solved recursively.

In [wan 93], J.L. Wang, et al considered a two queue priority system, where real time traffic is given high priority over non-real time traffic. The probability distributions for inter-departure times of cells from each queue are obtained.

In order to overcome the difficulties in output characterization of ATM switch, almost all the call admission control procedures and performance analyses reported in the literature, assume "Node Decomposition". To use this to determine whether to accept a call, intermediate switches in the path use the call characteristics as they appear at source; this in effect assumes that the characteristics will not be effected by the upstream switches. However there has been little research, (except [lau 93]) in validating this assumption. In [lau 93], the authors attempted the problem of validation of "Nodal decomposition" approach through extensive simulations. Both homogeneous as well as heterogeneous ON-OFF sources are considered to study the input-to-output distortion in individual traffic source as a function of peak rate of each source and overall load factor. The authors also studied the cross-correlations amongs the output sources. This paper summararily reports two conditions under which nodal decomposition can be applied in network-wide performance modeling. These are – 1. *If the peak access rate of each source does not exceed 5 % of the total link capacity, source distortion will be negligible.* 2. *Should no more than 10 % of the departing sources go to the same immediate downstream link, inter source cross-correlation will have negligible effects on the queuing performance of the downstream nodes.*

From the congestion control and call management points of view, burstiness is one of the important properties of traffic whose knowledge enables us in designing call admission control procedures with better utilization of

buffer and bandwidth resources. Friesen and Wong [frie 93] considered interconnection of user nodes which are fed by multiple traffic sources and switch nodes. In the presence of bursty traffic, they analyzed mean queue lengths, mean delays at every user node and switch node. It is observed that mean queue length and mean delay are larger at the user node than at the first switch node. Similarly, the average queue length and mean delay are larger at the first switch node than at the second switch. Smoothing or Burst reduction of bursty sources is claimed as the reason for this. The smoothing effect increases for higher load. S.Low and P.Varaiya [low 91, low 93] defined burstiness of traffic in terms of the buffer required at the server for the given service rate. Using a deterministic fluid flow model, they show that both fixed rate and leaky bucket servers are burst reducing.

We consider a queue with N ON-OFF sources as input, served by a slotted channel. Given the characteristics of each call at the input side, we obtain expressions for the density function of its burst length at the output side. The inter-departure time between cells of a call within a burst, and hence the length of a burst at the output side, depends not only on the instantaneous queue length, but also on the instantaneous states of all the calls at the input side. This is modeled using the Maximum Entropy principle. The queue length distribution can be obtained by approximating the multiplexed traffic at the input to the queue as a 2- state Markov Modulated Poisson Process (MMPP) [hef 86].

The problem attempted in this paper is different from the earlier literature [mat 94-I, mat 94-II, ohba 91, stav 91, wan 93]. W. Matragi, et al [mat 94-I, mat 94-II] and Ohba, et al [ohba 91] considered only GI- stream in the presence of batch traffic. I. Stavarakakis [stav 91] and J.L. Wang, et al [wan 93] considered only combined output characteristics. The output characterization of individual ON-OFF sources is considered important for obvious applications in telephone, data networks, etc. Also to the best knowledge of the authors, usage of the Maximum Entropy principle for estimation of the service time density function is new.

In this paper, we analyze the distribution of burst length of the tagged ON-OFF source at the output of a multiplexer with infinite buffer. The input to the multiplexer is a set of heterogeneous or homogeneous ON-OFF sources. Section 2 presents the model as an infinite buffered queue fed by arrivals from a number of ON-OFF sources. The effect of other sources on the output characteristics of the tagged call is twofold. The inter-cell departure time of two successive cells within a burst of the tagged call depends on the number of sources that are in ON state at that instant. Section 3 introduces the notion of instantaneous bandwidth available to the tagged call which models the number of sources that are in ON state at that instant. We also present in this section the usage of Maximum Entropy principle to estimate the density function of the instantaneous bandwidth. The inter-cell departure time of successive cells within a burst of the tagged call also depends on the queue length distribution which in turn depends on the state of other sources. In Section 4, a modified queue model with variable server is presented. The input to this queue is cells from the tagged call. The variable service time of the server is to model the instantaneous bandwidth available to the tagged call. Also the server is assumed to go on vacation at the beginning of ON state which will model the dependence in the queue length distribution. In Section 5, density function of the output burst length is analysed. Some numerical examples are presented in Section 6, and compared with simulation results. Section 7 gives the concluding remarks.

2 MODEL DESCRIPTION

We consider an ATM statistical multiplexer with an infinite buffer serving N ON-OFF sources each generating cells of constant size. The multiplexer is served by a single channel with capacity C bits/sec. The channel is slotted with slot size equal to the service time of a cell. This multiplexer buffer can be modeled as a discrete-time single server system.

Each ON-OFF source alternates between ON and OFF states. During the ON state, source i ($i = 1, \dots, N$) generates traffic at a constant rate R_i bits/sec. Without loss of generality, we consider the size of a cell to be 53 bytes (ATM standard). Each source is modeled as a discrete source such that it can be described completely at time instants $\tau_0, \tau_1, \dots, \tau_{j-1}, \tau_j, \tau_{j+1}, \dots$, where $a_i \equiv \tau_{n-1} - \tau_n = 53 \times 8 / R_i$ sec., for all n . At an arbitrary instant τ_n ,

if the source i is in ON state, the source will continue to be in the ON state at time instant τ_{n+1} with probability α_i and with probability $(1 - \alpha_i)$, the source will switch to OFF state at τ_{n+1} . Similarly if the source is in OFF state at the instant τ_n , it will continue to be in OFF state at the instant τ_{n+1} with probability β_i and switch to ON state with probability $(1 - \beta_i)$. The source will emit a cell of size 53 bytes at the time instant τ_n , if it is in the ON state at that instant. Let θ_{ON}^i be the average number of cells emitted by source i during an ON period and θ_{OFF}^i be the average length of OFF period in units of cell times. Then we get

$$\theta_{\text{ON}}^i = \frac{1}{1 - \alpha_i}, \quad \theta_{\text{OFF}}^i = \frac{1}{1 - \beta_i}$$

So the average traffic load of source i , R_{avg}^i is given by,

$$R_{\text{avg}}^i = \frac{\theta_{\text{ON}}^i}{\theta_{\text{ON}}^i + \theta_{\text{OFF}}^i}$$

Consider now, the intercell-departure time for cells belonging to the same ON period of source i . This intercell-departure time depends not only on the number of cells belonging to other sources, served in between two cells of source i but also on the queue length at the departure time of the first cell of the tagged cell pair. The number of cells belonging to other sources, that are served in between the tagged cell pair, is a random variable and depends on the number of other sources that are in the ON state at that instant. Using this, the statistical multiplexer can be approximated as an infinite buffered queue (with only the tagged ON-OFF source i as the input), which is being served by a server with a random service rate u ; the server is also assumed to go on vacation before starting service to a cell. Thus the server with a variable service rate takes into account the fact that the effective instantaneous bandwidth available to the cells of the tagged source is variable and depends on the number of ON-OFF sources that are in ON state at that instant. The vacation period of the server is also a random variable and takes into account the fact that before commencement of service to a cell, the cells that are waiting in the multiplexer, need to be served.

3 INSTANTANEOUS BANDWIDTH

In the statistical multiplexer, we consider the service of cells belonging to the same ON state of source i . Specifically, between two cells of the same ON state of source i , depending on the states of other sources, cells belong to other sources will also get served. If the number of cells belonging to other sources present in between two cells belonging to the source i is large enough so that the service time for all those cells is more than a_i , then the instantaneous inter-departure time between the cells of source i is more than a_i and is equal to the total service time of the cells that are queued in between those two cells. If this number is small enough so that the total service time for all those cells is smaller than a_i , then there are two possible cases:

- If the next cell of source i has arrived before the departure of the previous cell, then the interdeparture time between the cells of source i will be equal to the service time of cells belonging to the other sources.
- Otherwise, the inter-departure time between the cells of source i will be equal to a_i .

The number of cells of other sources arriving between the cells of source i depends on which sources are ON at that instant and their peak rates. Consider a specific situation when all N sources are in the ON state. Here in between two cells belonging to source i , the average number of cells of other sources, that are queued up, can be calculated as follows:

The average number of cells belonging to other sources

$$\begin{aligned}
 &= \frac{R_1}{R_i} + \frac{R_2}{R_i} + \dots + \frac{R_{i-1}}{R_i} + \frac{R_{i+1}}{R_i} + \dots + \frac{R_N}{R_i} \\
 &= \frac{1}{R_i} \sum_{j=1, j \neq i}^N R_j
 \end{aligned}$$

The service time required to serve these cells

$$= \frac{1}{R_i} \left[\sum_{j=1, j \neq i}^N R_j \right] \frac{53 \times 8}{C}$$

So, the average inter-departure time between the cells belonging to source i

$$\begin{aligned}
 &= \frac{53 \times 8}{C} \left[1 + \frac{1}{R_i} \sum_{j=1, j \neq i}^N R_j \right] \\
 &= \frac{53 \times 8}{u_i}
 \end{aligned}$$

where $u_i = \frac{C R_i}{\sum_{j=1}^N R_j}$ is the instantaneous channel bandwidth of source i .

Consider another situation where only source i is ON. Then no other cells will be queued in between two cells of source i . In this case, the instantaneous channel bandwidth of source i , will be $u_i = C$.

The instantaneous bandwidth u_i available to source i can be defined as the state of the system with respect to source i , and is a discrete random variable which can take upto (2^{N-1}) values. Analysis involving a discrete random variable with such a large state space may not be practical. When the rate $\frac{C}{R_j}$, for all j , is large enough, the instantaneous bandwidth, u_i can be approximated to be a continuous random variable. If the distribution of u_i , is known, it means that the effect of all other sources on source i has been characterized.

Approximating a discrete random variable as a continuous random variable involves obtaining density function of a continuous random variable with point probabilities as constraints. In principle, this can be formulated as a Maximum Entropy problem with the density function as the optimizing variable and the point probabilities of the discrete random variable as the constraints. Due to the large size of the constraint set, this problem is complex; we simplify this by considering only a fixed and small set of constraints.

3.1 Maximum Entropy Principle

Consider the instantaneous bandwidth available to source i as the state of the system. Dropping the subscript i , we denote this as u . Assume that we know only its minimum, maximum and the average. Given this, the Maximum Entropy principle [shor 80, jay 57, wil 70, fer 70, kou 94], can be used to estimate the density function of u . For the last four decades, Maximum Entropy principle is being used in various engineering fields like Operation Research, Transportation, Queueing theory, etc for estimation of the state probability distribution in the absence of complete information about the state of the system. Of late Maximum Entropy principle found

applications in the area of ATM networks as well. Kouvatso, et al [kou 94] has used Maximum Entropy principle for estimating the queue length distribution of the statistical multiplexer. In this paper, we use Maximum Entropy principle to estimate density function of the instantaneous bandwidth available when we know only its average. The Maximum Entropy principle will "choose" the density function such that the entropy is maximized with the given information as constraints. In other words, if $p(u)$ is the density function of u , we find $p(u)$ by maximizing

$$\text{Entropy, } H(u) = - \int_{u_1}^{u_2} p(u) \ln p(u) du \quad (1)$$

such that,

$$\int_{u_1}^{u_2} p(u) du = 1 \quad (2)$$

$$\int_{u_1}^{u_2} up(u) du = \bar{u} \quad (3)$$

where,

u_1 = Minimum value of that u can attain

u_2 = Maximum value of that u can attain

Here $u_2 = C$, channel capacity.

\bar{u} = Average of u

Clearly, $p(u)$, that can be obtained from above, may not be true density function of u . Also the available information about u may not be sufficient to obtain the actual density function of u . Maximum Entropy principle will estimate the density function which satisfies the given information, but mostly non-committal about whatever not known. Also if we re-estimate the density function with additional information, the so-obtained density function may be different from that obtained previously. Hence the density function obtained from this Maximum Entropy principle will be an approximation to the true density of the system state u .

The solution for the above set of equations is discussed in Appendix and is given by,

$$p(u) = e^{\lambda_1 - 1} e^{\lambda_2 u}$$

where λ_1, λ_2 can be obtained from,

$$e^{\lambda_1 - 1} (e^{\lambda_2 u_2} - e^{\lambda_2 u_1}) = \lambda_2 \quad (4)$$

$$\frac{u_2 e^{\lambda_2 u_2} - u_1 e^{\lambda_2 u_1}}{e^{\lambda_2 u_2} - e^{\lambda_2 u_1}} - \frac{1}{\lambda_2} = \bar{u} \quad (5)$$

It is argued in the Appendix that Eq. (5) has unique solution for λ_2 . It can also be observed that for moderate to high load conditions ($0.4 \leq \rho \leq 0.99$), where the average instantaneous bandwidth $\bar{u} \leq \frac{u_1 + u_2}{2}$, the solution λ_2 is -ve. Since, as reported in the literature, at low load conditions, the input-output distortion is negligible, we consider here only the case, $\lambda_2 \leq 0$.

Now rewriting Eq. (5), we get

$$\frac{u_2 - u_1}{u_2 - \bar{u} - \frac{1}{\lambda_2}} = 1 - e^{\lambda_2(u_2 - u_1)} \quad (6)$$

Since λ_2 is -ve,

$$e^{\lambda_2(u_2-u_1)} \leq 1 \quad (7)$$

Assuming that the term, $e^{\lambda_2(u_2-u_1)}$ in Eq. (6) is negligibly small, the solution for Eq. (6) is given by –

$$\lambda_2 \approx \frac{1}{u_1 - \bar{u}} \quad (8)$$

It is observed that the above assumption is valid with varying accuracies in many examples we considered. The ratio of channel capacity and peak rate of the call is one of the factors which effect the validity of the assumption. Although, exact condition for the validity is yet to be derived, an empirical condition can be arrived at by conditioning that $e^{\lambda_2(u_2-u_1)}$ is negligible.

For $e^{\lambda_2(u_2-u_1)}$ to be negligibly small,

$$|\lambda_2(u_2 - u_1)| \geq 10$$

But in this case, λ_2 is given by Eq. (8). Hence the empirical condition for the validity of the above assumption is given by –

$$\left| \frac{u_2 - u_1}{u_1 - \bar{u}} \right| \geq 10 \quad (9)$$

In case the above condition is not satisfied, Eq (5) can be solved numerically for λ_2 . The following successive approximation algorithm is used to evaluate λ_2 iteratively with initial guess is given by Eq. (8). The main advantage of this algorithm is its insensitivity to the initial guess. The necessary condition for this algorithm to converge is given by –

$$e^{\lambda_2(u_2-u_1)} \leq \left(\frac{u_1 - \bar{u}}{u_2 - \bar{u}} \right)^2$$

which can be satisfied for all moderate to heavy load conditions.

So

$$u_1 = \frac{C.R_i}{\sum_{j=1}^N R_j}$$

Similarly maximum, u_2 of the state of the system is the maximum possible share of the channel bandwidth for source i as it occurs when the instantaneous load is minimum possible, i.e. all the sources, except the tagged source i , are in OFF state.

Let $\nu = (x_1, x_2, \dots, x_{i-1}, 1, x_{i+1}, \dots, x_N)$ be the combined state of all sources given that the source i is in ON state, where

$$x_j = \begin{cases} 0 & \text{if the source } j \text{ is in OFF state} \\ 1 & \text{Otherwise} \end{cases}$$

Also let $u_i(\nu)$ and $p(\nu)$ be instantaneous bandwidth available for source i when the sources are in state ν and joint probability that the sources are in state ν , respectively. Then the expected value of instantaneous bandwidth available, \bar{u} for source i when it is ON state can be obtained as –

$$\bar{u} = \sum u_i(\nu) \cdot p(\nu)$$

Since all the sources are independent of each other, we can write,

$$p(\nu) = \prod_{j=1, j \neq i}^N p(x_j)$$

where $p(x_j)$ is the probability that source j is in ON state, if $x_j = 1$ or in OFF state if $x_j = 0$. Also it can be easily shown that

$$p(x_j = 1) = \frac{(1 - \beta_j)}{(1 - \alpha_j) + (1 - \beta_j)}$$

and

$$p(x_j = 0) = \frac{(1 - \alpha_j)}{(1 - \alpha_j) + (1 - \beta_j)}$$

4 INTER-CELL DEPARTURE TIME

We now consider the infinite buffer queue served by a single server with capacity, u where u is a random variable with density function $p(u)$. The customers to this queue are the cells belonging to source i . We also assume that the server will be on vacation at the time of arrival of each cell. The vacation period is a random variable, v (≥ 0). Let $b = \frac{53 \times 8}{u}$ be the service time of a cell in this queueing system, with $f_b(b)$ and $B^*(s)$ as the density function and Laplace Transform of b respectively.

The vacation period seen by an arriving cell of source i will be the time required to serve the cells that are waiting in queue at the arrival instant of this cell. The inter-cell departure time at the output of this queue is equivalent to the inter-departure time of cells belonging to source i , from this multiplexer.

We consider two cases. When the vacation period for the cell is so large that before the start of service of this cell, next cell of the same ON period (or burst) has arrived into the queue, then the instantaneous inter-cell departure time between the present cell and the next, is equal to the service time of the cell in the above queueing system. Let us define d_1 as inter-departure time given that new cell has arrived before the service of previous cell started. Then $d_1 = b$ and f_{d_1} and $D_1^*(s)$ are the density function and Laplace Transform for d_1 , respectively.

In the other situation, the vacation is small enough so that the service of the cell starts before the arrival of the next cell of the same burst. Let us define d_2 as the inter-departure time between the cells in this case. Then we get $d_2 = \max(b, a)$, where a is the interarrival time of cells of source i within a burst (subscript i removed for simplification).

Since d_2 is random variable, let us define $f_{d_2}(d)$ and $D_2^*(s)$ as the density function and Laplace Transform of d_2 , respectively. This yields

$$f_{d_2}(d) = f_b(b)F_a(d) + F_b(d)f_a(d)$$

where, $F_b(\cdot)$ is the distribution function of b and $f_a(\cdot)$ and $F_a(\cdot)$ are the density and distribution functions of a , respectively.

Since a is constant,

$$f_a(d) = \delta(d - a)$$

where δ is dirac delta function.

$$F_a(d) = \begin{cases} 1 & \text{if } d \geq a \\ 0 & \text{otherwise} \end{cases}$$

So rewriting,

$$f_{d_2} = f_b(d)F_a(d) + F_b(d)\delta(d - a)$$

Then

$$\begin{aligned} D_2^*(s) &= \int_0^\infty f_{d_2}(d)e^{-ds}dd \\ &= \int_0^\infty f_b(b)e^{-bs}db + F_b(a)e^{-as} \end{aligned}$$

4.1 Vacation Period And Queue Length Distribution

In the previous section, we considered the server with vacations, where the vacation period is equivalent to the service time required to serve all the cells ahead of tagged cell of the tagged source in the multiplexer buffer. The vacation period at any arbitrary time instant is the time required to serve a cell at the channel rate C times the queue length at that instant. Therefore, the queue length distribution of the multiplexer will be needed to find the vacation period distribution.

The statistical multiplexing of N ON-OFF sources can be approximated to be a 2-state MMPP as proposed in Heffes, et al [hef 86]. The queue length distribution of the $MMPP | D | 1$ infinite buffer queue may be obtained as proposed by Ramaswami [ram 80, ram 88] and Lucantoni [luc 91]. We define q as the queue length of the multiplexer at the cell departure instants and $q(n)$ is the steady state queue length distribution.

Consider the probability that at the start of the service time of a cell of source i in the multiplexer, the next cell of the same ON state is also waiting. Consider the instant when the service of a cell belonging to source i has started. Let the queue length at that instant be given by q' , with distribution, $q'(n) = q(n - 1)$, for $n = 1, 2, \dots$

Let S be the set of all possible combined states of all sources and R_ν be the total arrival rate of cells into the multiplexer when the combined state is ν , i.e

$$R_\nu = \sum_{j=1}^N x_j R_j, \quad x_j \in S$$

Between two cell arrivals of source i , the number of cells of other sources that can arrive is given by $n_\nu = aR_\nu$.

If the queue length q' is greater than n_ν when service to a cell of source i starts then another cell of the same ON state is also waiting. Let p_ν denote the probability of this event given that the combined state of all sources is ν .

$$p_\nu = \sum_{n=n_\nu+1}^{\infty} q'(n)$$

Since n_ν is real number, it can be written as –

$$n_\nu = n_I + n_f = n_I(1 - n_f) + (n_I + 1)n_f$$

where n_I and n_f are integral and fractional part of n_ν , respectively. Then

$$p_\nu = (1 - n_f) \times q'(n_I + 1) + \sum_{n=n_I+2}^{\infty} q'(n)$$

Let p denote the probability averaged over state ν that at the start of service of a cell belonging to source i , the next cell of the same ON state has also arrived. Then

$$p = \sum_{\nu \in S} p_\nu p(\nu)$$

Now let us define d as the inter-departure time of cells of the same ON state of source i . Then d is given by,

$$d = d_1 p + (1 - p) d_2$$

Let $f_d(d)$ and $D^*(s)$ be the density function and Laplace Transform of d , respectively where $D^*(s)$ is given by

$$D^*(s) = D_1^*(sp) D_2^*(s(1 - p))$$

5 BURST LENGTH AT THE OUTPUT

We define the burst length of source i at the output of the multiplexer as the time difference between the start of service of first cell of the burst at the input to the departure of the last cell of that burst. Let br_n denote the burst length at the output when there are n cells in the corresponding burst at the input side. Assuming that within an ON state of tagged source, variations in both the instantaneous channel bandwidth as well as vacation periods are negligible, br_n can be approximated as

$$br_n = (n - 1)d$$

if $Br_n^*(s)$ is the Laplace transform of br_n ,

$$Br_n^*(s) = D_1^*((n - 1)ps) D_2^*((1 - p)(n - 1)s) \quad (10)$$

But the probability that there are n cells in the burst at the input is $\alpha^{n-1}(1-\alpha)$, for $n = 1, 2, \dots$. Let br denote burst length at the output averaged over n and $Br^*(s)$ is the Laplace transform of br ; where

$$br = \sum_{n=1}^{\infty} \alpha^{n-1}(1-\alpha)br_n$$

Then

$$Br^*(s) = \prod_{n=1}^{\infty} Br_n^*(\alpha_{n-1}(1-\alpha)s) \quad (11)$$

5.1 Average of Burst Length

Differentiating Eq. (11) w.r.t. s ,

$$Br^{*'}(s) = \sum_{j=1}^{\infty} Br_j^{*'}(\alpha^{j-1}(1-\alpha)s) \alpha^{j-1}(1-\alpha) \prod_{i=1, i \neq j}^{\infty} Br_i^*(\alpha^{i-1}(1-\alpha)s)$$

Substituting $s = 0$, we get

$$Br^{*'}(0) = (1-\alpha) \sum_{j=1}^{\infty} \alpha^{j-1} Br_j^{*'}(0) \quad (12)$$

Differentiating Eq. (10) w.r.t. s , and substituting $s = 0$, we get

$$Br_n^{*'}(0) = (n-1)pD_1^{*'}(0) + (1-p)(n-1)D_2^{*'}(0) \quad (13)$$

Substituting Eq. (13) in Eq. (12),

$$Br^{*'}(0) = (1-\alpha) \sum_{j=1}^{infy} \alpha^{j-1}(j-1) [pD_1^{*'}(0) + (1-p)D_2^{*'}(0)]$$

Now

$$D_1^{*'}(s) = B^{*'}(s) = - \int_0^{\infty} b f_b(b) e^{-bs} db$$

Substituting $s = 0$ in the above equation, we get

$$\begin{aligned} D_1^{*'}(0) &= - \int_0^{\infty} b f_b(b) db \\ &= - \int_{u_1}^{u_2} \frac{53 \times 8}{u} \frac{\lambda_2}{[e^{\lambda_2 u_2} - e^{\lambda_2 u_1}]} e^{\lambda_2 u} du \end{aligned}$$

Similarly,

$$\begin{aligned} D_2^{*'}(s) &= - \int_a^\infty b f_b(b) e^{-bs} db - a F_b(a) e^{-as} \\ &= - \int_{u_1}^{R_1} \frac{53 \times 8}{u} \frac{\lambda_2}{[e^{\lambda_2 u_2} - e^{\lambda_2 u_1}]} e^{\lambda_2 u} du - a F_b(a) \end{aligned}$$

The average of burst length, \overline{br}

$$= (\alpha - 1) \sum_{j=1}^{\infty} \alpha^{j-1} (j - 1) [p D_1^{*'}(0) + (1 - p) D_2^{*'}(0)] \quad (14)$$

where,

$$D_1^{*'}(0) = - \int_{u_1}^{u_2} \frac{53 \times 8}{u} \frac{\lambda_2}{[e^{\lambda_2 u_2} - e^{\lambda_2 u_1}]} e^{\lambda_2 u} du \quad (15)$$

$$D_2^{*'}(0) = - \int_{u_1}^{R_1} \frac{53 \times 8}{u} \frac{\lambda_2}{[e^{\lambda_2 u_2} - e^{\lambda_2 u_1}]} e^{\lambda_2 u} du - a F_b(a) \quad (16)$$

6 NUMERICAL RESULTS AND DISCUSSION

In this section, we discuss two set of numerical experiments that were made to gauge the accuracy of the expressions derived in previous sections. The average burst length at the output side of the multiplexer is calculated and compared with simulation results. In these two experiments, we consider channel bandwidth of 155 Mbits/sec and cell size of 53 bytes. In both the experiments, calls with same characteristics (i.e. homogeneous calls) were considered.

In the first experiment, each call is described by, Peak Rate, $R_i = 20$ Mbits/sec., $\alpha_i = 0.95$ and Average/Peak Rate ratio = 0.4576. The experiment was conducted with 3 different load factors, where ρ is defined as

$$\rho = \frac{C}{\sum_{i=1}^N R_i}$$

No. of Calls	Load Factor ρ	Burst Length at the output in sec.		
		Simulation	Calculated with QLDs from Sim.	Calculated with QLDs from appr.
13	0.767	410	411	428
15	0.8856	428	438	456
16	0.944	434	454	468

In the second experiment, we consider each call with Peak Rate, $R_1 = 10$ MBits/sec., $\alpha_1 = 0.95$ and Average/Peak rate ratio = 0.4576.

No. of Calls	Load Factor ρ	Burst Length at the output in sec.		
		Simulation	Calculated with QLDs from Sim.	Calculated with QLDs from appr.
25	0.738	812	828	848
27	0.797	814	858	883
30	0.8856	824	896	934

It can be observed from above tables that the percentage of error in the burst length calculated with Queue Length Distribution (QLD) obtained from 2- state MMPP approximation is more than in those calculated using the QLDs obtained from the simulations. This may be due to fact that the probabilities of higher order queue lengths are underestimated in the 2- state MMPP approximation. The QLDs calculated with approximations using higher order MMPP (MMPP with more than 2 states) may improve the percentage of error.

It is also observed that as the number of calls increases, the percentage of error in burst lengths also increases. This may be attributed to the loss of information about the higher order moments of instantaneous bandwidth available to the tagged source in the Maximum Entropy approximation. To be more clear, let us consider "occupied channel bandwidth" in the homogeneous case, which is the sum of the peak rates of those sources which are in ON state. The occupied channel bandwidth is a random variable which depends on another random variable, number of sources that are in ON state at that instant. The second moment of the occupied channel bandwidth depends on the second moment of the number of sources that are in ON state whose dependence on the total number of sources is second order polynomial. So any increase in the total number of sources, would cause the second moment of the occupied channel bandwidth to increase by a second order polynomial factor. Hence as the total number of sources increases, the difference between the exact second moment and the estimated second moment from Maximum Entropy principle with only first moment as the constraint, increases by a second order polynomial factor. Since the occupied channel bandwidth and instantaneous bandwidth available to the tagged call are closely related, same arguments holds good for instantaneous bandwidth as well. Hence as the number of sources increases, the loss of information about higher order moments is higher in the Maximum Entropy approach.

7 CONCLUSION

An approximate expression is derived for the burst length of a tagged call at the output of an ATM switch by approximating the statistical multiplexer as a single variable server infinite buffer queuing system with only cells from the tagged call as customers. Each incoming cell also sees the server in randomly variable vacation periods. The density function of the service rate of the server is approximated using Maximum Entropy Principle. Two numerical examples are presented to gauge the accuracy/inaccuracy of the approximation. Considering the fact that only first moment is used as constraint, the accuracy of the results is impressive. In the authors' opinion, the main contributions of the paper are 1) introduction of Maximum Entropy principle for the estimation service time density function and 2) modeling of the statistical multiplexer as a variable server queuing systems with server vacations. This approach can be extended further by including more constraints for better estimation of the density function of the instantaneous bandwidth available.

APPENDIX 1 SOLUTION OF EQ. (1), EQ. (2) AND EQ. (3):

Using Langrangian principle,

$$F(p) = - \int_{u_1}^{u_2} p(u) \ln p(u) du + \lambda_1 \left[\int_{u_1}^{u_2} p(u) du - 1 \right] + \lambda_2 \left[\int_{u_1}^{u_2} u p(u) du - \bar{u} \right] \quad (17)$$

Where λ_1, λ_2 are Langrangian Coefficients.

Differentiate Eq. (17) with respect to $p(u)$ and equate it to 0.

$$\frac{dF}{dp} = - \int_{u_1}^{u_2} \left[\ln p(u) du + p(u) \cdot \frac{1}{p(u)} du \right] + \lambda_1 \left[\int_{u_1}^{u_2} du \right] + \lambda_2 \left[\int_{u_1}^{u_2} u du \right] = 0$$

Rewriting,

$$\int_{u_1}^{u_2} [-\ln p(u) - 1 + \lambda_1 + \lambda_2 u] du = 0$$

After simplification, We get

$$p(u) = e^{\lambda_1 - 1} \cdot e^{\lambda_2 u} \quad (18)$$

Substituting Eq. (18) in Eq. (2),

$$\int_{u_1}^{u_2} e^{\lambda_1 - 1} \cdot e^{\lambda_2 u} du = 1$$

We get,

$$e^{\lambda_1 - 1} (e^{\lambda_2 u_2} - e^{\lambda_2 u_1}) = \lambda_2 \quad (19)$$

Substituting Eq. (18) in Eq. (3),

$$\int_{u_1}^{u_2} u e^{\lambda_1 - 1} \cdot e^{\lambda_2 u} du = \bar{u}$$

After simplification and substituting Eq. (19), we get,

$$\frac{u_2 e^{\lambda_2 u_2} - u_1 e^{\lambda_2 u_1}}{e^{\lambda_2 u_2} - e^{\lambda_2 u_1}} - \frac{1}{\lambda_2} = \bar{u}$$

So the probability density function estimated from Maximum Entropy principle is given by,

$$p(u) = e^{\lambda_1 - 1} e^{\lambda_2 u}$$

Where λ_1, λ_2 can be obtained from,

$$e^{\lambda_1 - 1} (e^{\lambda_2 u_2} - e^{\lambda_2 u_1}) = \lambda_2$$

$$\frac{u_2 e^{\lambda_2 u_2} - u_1 e^{\lambda_2 u_1}}{e^{\lambda_2 u_2} - e^{\lambda_2 u_1}} - \frac{1}{\lambda_2} = \bar{u} \quad (20)$$

Evaluation of Lagrangian Coefficients

Define,

$$f = \frac{u_2 e^{\lambda_2 u_2} - u_1 e^{\lambda_2 u_1}}{e^{\lambda_2 u_2} - e^{\lambda_2 u_1}} - \frac{1}{\lambda_2} \quad (21)$$

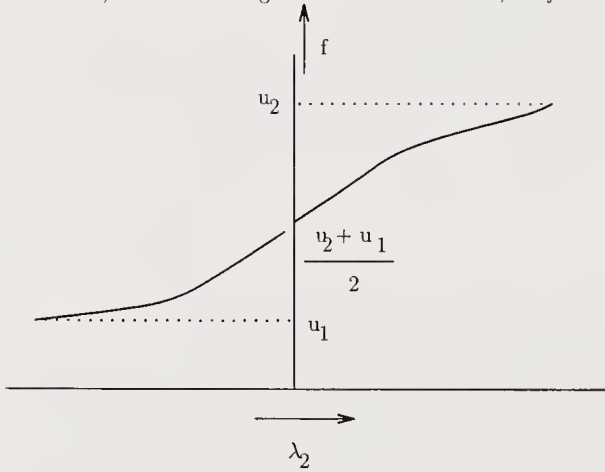
We can easily show that -

$$\lim_{\lambda_2 \rightarrow 0} f = \frac{u_1 + u_2}{2}$$

$$\lim_{\lambda_2 \rightarrow -\infty} f = u_1$$

$$\lim_{\lambda_2 \rightarrow +\infty} f = u_2$$

The curve, obtained through numerical simulations, for f as a function of λ_2 is shown in the figure below.



From the above figure and the limits of f , we can conclude that Eq. (20) has unique solution λ_2 .

REFERENCES

- [ohba 91] Y. Ohba, M. Murata, H. Miyahar, *Analysis of Interdeparture Processes for Bursty Traffic in ATM networks*, IEEE J-SAC, April 1991, Vol. 9, No. 3, P.No. 468-476.
- [stav 91] I.Stavarakakis, *Efficient Modeling of Merging and Splitting processes in Large Networking Structures*, IEEE J-SAC, Oct 1991, Vol. 9, No. 8, P. No. 1336-1347.
- [frie 93] V.J.Friesen, J.W.Wong, *The Effect of Multiplexing, Switching and Other Factors on the Performance of Broadband Networks*, Proc. of IEEE INFOCOM '93, P.No. 1194-1203.
- [low 91] S. Low, P. Varaiya, *A Simple Theory of Traffic and Resource Allocation in ATM*, Proc. of IEEE GLOBE-COM '91, P.No. 1633-1637.
- [low 93] S. Low, P. Varaiya, *Burstiness Bounds for Some Burst Reducing Servers*, Proc. of IEEE INFOCOM '93, P.No. 1a.1.1-1a.1.8.
- [mat 94-I] W. Matragi, C. Bisdikian, K. Sohraby, *Jitter Calculus in ATM Networks; Single Node Case*, Proc. of IEEE INFOCOM '94, P.No. 232-241.
- [mat 94-II] W. Matragi, C. Bisdikian, K. Sohraby, *Jitter Calculus in ATM Networks; Multi Node Case*, Proc. of IEEE INFOCOM '94, P.No. 242-251.
- [rob 92] J.Roberts, F. Guillemin, *Jitter in ATM Networks and its Impact on Peak Rate Enforcement*, Performance Evaluation, 1992, Vol. 16, P.No. 35-48.
- [boy 92] P.E.Boyer, F.M.Guillemin, M.J. Servel, J.P.Coudreuse, *Spacing Cells Protects and Enhances Utilization of ATM Network Links*, IEEE Network, Sep 1992, P.No. 38-49.
- [cid 94] I. Cidon, A. Khamisy, M. Sidi, *Dispersed Messages in Discrete-Time Queues: Delay, Jitter and threshold Crossing*, Proc. of IEEE INFOCOM '94, P.No. 218-223.
- [wan 93] J.L. Wang, J.P. Zhou, C.Wang, Y.H. Fan, *Interdeparture processes of Traffic from ATM Networks*, Proc. of IEEE INFOCOM '93, P.No. 1337-1341.
- [hef 86] H. Heffes, D.M. Lucantoni, *A Markov Modulated Characterization of Packetized Voice and Data Traffic and Related Statistical Multiplexer Performance*, IEEE J-SAC, Sep 1986, Vol. SAC-4, No. 6. P. No. 856-868.
- [shor 80] J.E. Shore, R.W. Johnson, *Axiomatic Derivation of the Principle of Maximum Entropy and the Principle of Minimum Cross-Entropy*, IEEE Trans. on Info. Theory, Jan 1980, Vol. IT-26, No. 1, P. No. 26-36.
- [jay 57] E.T. Jaynes, *Information Theory and Statistical Mechanics*, Physical Review, May 1957, Vol. 106, No. 4, P.No. 620-630.
- [wil 70] A.G.Wilson, *The Use of the Concept of Entropy in System Modeling*, Operational Res. Quarterly, Vol. 21, 1970, No. 2, P.No. 247-265.
- [fer 70] A.E. Ferdinand, *A Statistical Mechanical Approach to System Analysis*, IBM J. Research, 1970, P.No 539-547.
- [kou 94] D.D. Kouvatsos, N.M. Tabet-Aouel, S.G. Denazis, *ME-based Approxiamtions for Genearl Discrete-time Queueing Models*, Performance Eval., 1994, Vol. 21, P.No. 81-109.
- [ram 80] V. Ramaswami, *The N/G/1 Queue and its Detailed Analysis*, Adv. Appl. Probability, 1980, Vol. 12, P.No. 222-261.
- [ram 88] V. Ramaswami, *A Stable Recursion for the Steady State Vector in Markov Chains of M/G/1 Type*, Comm. Stat.- Stochastic Models, 1988, Vol. 4(1), P.No. 183-188.
- [luc 91] D.M. Lucantoni, *New Results on the Single Server Queue with a Batch Markovian Arrival Process*, Comm. Stat.- Stochastic Models, 1991, Vol. 7(1), P.No. 1-46.
- [lau 93] Wing-cheong Lau, San-qi Li, *Traffic Analysis in Large-Scale High-Speed Integrated Networks: Validation of Nodal Decomposition Approach*, Proceedings of IEEE INFOCOM '93, 1993.

BIOGRAPHY

Bose, Sanjay K.: Prof. Bose did his Ph.D. from the State University of New York, Stobny Brook in 1980. He was with the Corporate RD of the General Electric Co. at Schenectady, N.Y during 1980-82. Since 1982 he has been on the faculty of the Indian Institute of Technology, Kanpur where he is currently a Professor in the

Department of Electrical Engineering. Prof. Bose has held visiting appointments at the University of Adelaide, Queensland University of Technology and the University of Pretoria. His research interests are in performance evaluation of computer and telecommunication networks. Prof. Bose is a member of Eta Kappa Nu, Sigma Xi and a Senior Member of IEEE.

Srivathsan K.R.: Prof. Srivathsan did his Ph.D. from Queen's University, Canada in 1981. He has been on the faculty of the Indian Institute of Technology, Kanpur since 1982 where he is currently a Professor in the Department of Electrical Engineering. He has been active in the area of Computer Networks and related applications and is one of the Coordinators of the ERNET project providing network facilities to academic and research institutions in India.

Using Markovian Models to Replicate Real ATM Traffics

Åke Arvidsson and Christer Lind^{†‡}*

University of Karlskrona/Ronneby and Telia Research AB[†]*

**Dept. of Telecommun. and Maths., Univ. of Karlskrona/Ronneby,
S-371 79 Karlskrona, Sweden. Email: akear@itm.hk-r.se*

Tel: +46 455 78053. Fax: +46 455 78057.

[†]Telia Research AB, Commun. Sys., Box 85,

S-201 20 Malmö, Sweden. Email: Christer.Lind@malmo.trab.se

Tel: +46 40 105137. Fax: +46 40 307029.

Abstract

Among the more commonly employed models for performance analysis of ATM networks, *e.g.* to dimension buffers in switches, we find Markov modulated Poisson processes (MMPPs) and Markov modulated Bernoulli processes (MMBPs). These models are often used with the only motivation that they are capable of producing bursty traffic. Although this is true in a general sense, little is known about whether that capability extends to the particular case of real traffics.

We report on an investigation where these models are tried in the latter sense. More precisely, we review and try a number of methods proposed for fitting MMPPs (MMBPs) to observed traffic data. The data consists of sixty traces which are extracted from the Bellcore Ethernet measurements according to length (short, medium, and long) and local average load (light and heavy). We then compare the performance of the buffer of a single server system when subject to the real traffic and the fitted model respectively.

It is found that the two cases differ significantly in terms of buffer occupancy, and that these differences are caused by deficiencies in the different fitting methods and possibly also by limitations in the models themselves. Nevertheless, some fitting methods are identified which, with further development, might work as models of burstiness within limited time spans on the order of two seconds. We also briefly comment the relationship between our results and recent works on fractal traffic characteristics.

Keywords

Bursty traffic model, ATM cell level traffic model, accuracy, Markov modulated Poisson Process, Markov modulated Bernoulli Process, MMPP, MMBP.

[‡]The major part of this work was carried out while Christer Lind was with the Department of Communication Systems, Lund Institute of Technology, Sweden.

1 MARKOVIAN MODELS FOR REAL ATM TRAFFICS

1.1 Markovian Models

Models of bursty traffic are frequently used in the context of performance analysis of ATM networks, *e.g.* to dimension buffers in switches. Among the more commonly employed models we find the Markov modulated Poisson processes (MMPPs) and Markov modulated Bernoulli processes (MMBPs). The two processes are doubly stochastic point processes where the rate of a Poisson (Bernoulli) process is governed by an underlying Markov chain in continuous (discrete) time. Arrivals and state transitions of the modulating chain are statistically independent. The processes are fully characterised by the number of states in the modulating chain s , the transition rates (probabilities) $q_{u,v}$, and the arrival rates (probabilities) r_u , $u, v \in \{1, \dots, s\}$.

To restrict the number of parameters, the number of states s is often set equal to two, in which case the model is referred to as a Switched Poisson (Bernoulli) Process, or an SPP (SBP). In the special case of the SPP (SBP) having an arrival rate of zero in one of its states, the process is called an Interrupted Poisson (Bernoulli) Process.

The main reasons why these models are frequently employed are probably their ability to match various burstiness characteristics, and their mathematical tractability. However, little is known about their actual relevance when it comes to producing a traffic that is not only generally bursty, but that in some meaning is equivalent to real traffic.

The current work is a preliminary attempt to investigate this aspect of simple Markovian models, typically SPPs and SBPs. The emphasis of the work is on their suitability for performance analysis, in particular with respect to buffer dimensioning. The general idea is to produce cell arrivals to an infinite buffer which is emptied by a single server, and study the number of cells present in the buffer at each arrival instant. A model that in our sense is equivalent to real traffic, would result in a buffer occupancy that is statistically identical to that of a real traffic.

To our knowledge, very few papers have been published where models are verified against real traffics in terms of buffer occupancy. Instead it appears that most researchers who verify models tend to do this against other models (!). One notable exception from this is the paper on video modelling published by Frater *et al.* (1994) and Rose (1994), where the queuing behaviour of a real traffic is compared to that of a model.

1.2 Replicating Real Traffics

Users wishing to establish a connection over an ATM network are required to declare a number of parameters characterising the traffic they wish to submit. These parameters include peak rate, sustainable rate, burst size, and possibly others. Typical factors affecting the choice of parameter settings include the nature of the application, characteristics of the user premises equipment, the access medium, and the tariff structures.

Testing models under this scenario, the model should represent the traffic actually submitted, *i.e.*, after possible shaping by the policing device. For a given trace, the user could declare virtually any set of parameters, and deliver the traffic in a number of conforming and non-conforming ways. To avoid restrictive presumptions regarding these parameters and delivery, we assume that the parameters are set such that the traffic can be passed transparently to the network, and therefore simply match the models directly to the traces.

The procedure of matching a model to a traffic trace is referred to as a fitting method. The fitting methods considered in this work can be classified in three categories: Sequence fitting, direct metrics fitting, and indirect metrics fitting. It is pointed out that the choice of modelling in discrete or continuous time is more a matter of mathematical convenience than of replication accuracy, Arvidsson *et al.* (1991).

Sequence Fitting

The idea of sequence fitting is based on the presumption that the trace is in fact produced by a specific model the parameters of which are unknown. Fitting a model to a trace therefore means to find the set of parameters of this particular model that have the highest likelihood of producing that sequence. The typical procedure is to start from an initial guess of the parameter set and successively improve it with respect to the likelihood of obtaining the trace until no further improvement can be obtained.

We have used two methods of this class, one due to Meier-Hellstern (1987) (KMH) and another one due to Rydén (1992) (TR). Both are developed for MMPPs with any number $s > 1$ of states, but are here applied to the case $s = 2$.

Direct Metrics Fitting

Direct metrics fitting does not presume that a certain model is actually valid, but simply aims at making the model in question reproduce certain “important” and mathematically tractable properties of the trace. Typical such properties fitted to are moments and correlations of inter arrival times and of the number of arrivals within intervals of length t .

We have considered four such methods, Rossiter (1987) (MR), Heffes *et al.* (1986) (HL), Gusella (1991) (RG), and Park *et al.* (1994) (DP). The three former are developed for and applied to MMPPs with $s = 2$ states and the latter to MMBPs with $s = 2$ states.

Indirect Metrics Fitting

Indirect metrics fitting means that the observed process is first transformed into another process which is then dealt with as for direct metrics fitting. The transformation procedure we have considered is the identification of “active periods” and “passive periods”, an idea first proposed by Jain *et al.* (1986). Active periods refer to uninterrupted sequences of one or more short inter arrival times, and passive ones to uninterrupted sequences of one or more long inter arrival times. Properties of interest in the transformed process include moments and correlations of the lengths of the two periods and of the activity within each of them.

We have used four such approaches, Solé *et al.* (1990) (SDG/1) and (SDG/4), Bonomi *et al.* (1994) (BMMP), and Lee *et al.* (1992) (LL). Both SDG/ x -methods refer to MMBPs with $s = 2$ states and allow for activities between zero and one during both periods. BMMP and LL refer to MMBPs and MMPPs with $s = 3$ and $s = 2$ states respectively, and both prescribe strictly no activity during passive periods and strictly full activity during active ones.

1.3 Preliminaries of the Investigation

It is well known that traffic characteristics depend heavily both on the source (*e.g.* video or data) and on the content (*e.g.* drama, sports, file transfer and www-retrievals). It is also clear that not even for a given source and content, there is such a thing as a “typical behaviour”. A general investigation of traffic replicating properties would therefore require tremendous amounts of recorded traffic traces. We have restricted ourselves to one class of traffic which could be labelled “LAN interconnect”. The motivation

for our particular choice is twofold: LAN interconnect is expected to be one of the first traffics to be sent over ATM, and LAN traffics measurements were readily available to us through the Bellcore (1989) measurements.

Numerous papers, *e.g.* Leland *et al.* (1994), Paxson *et al.* (1994), Pruthi (1995), and others, have reported on the self similar properties of these traffic traces, the presence of variations on all time scales, and the heavy tailed buffer occupancy distributions resulting from them. These findings raise fundamental questions regarding the relevance of Markovian models, in particular for those with small numbers of states s , the variabilities of which span a strictly limited time scale, *cf.* Andersen (1995).

The present work is, however, restricted to model variations within certain time scales. This is motivated by engineering aspects of buffer dimensioning, where loss constraints for slowly varying traffics may call for very large buffers, quite possibly large enough to violate delay constraints and even beyond reasonable physical limitations. (This becomes obvious when looking at buffer sizes and performance for systems that store excess traffic generated during working hours and transmit it during the nights.) Generally speaking, we can thus identify two kinds of variations: Fast variations which can be smoothed by a buffer, and slow variations which cannot. We are only interested in the former.

For slow variations we can see at least three possible ways: the first one is to multiplex a very large number of independent sources in which case even slow variations can be statistically multiplexed; the second one is to provide enough transmission capacity to handle the peaks and simply put up with the resulting poor utilisation in the valleys; and the third one is to trace the slow variations and dynamically adjust the allocated transmission capacity in accordance with the variations. Our work is based on the last approach, and we presume the presence of a control mechanism that dynamically adjusts the capacity of the server to the long term average of the traffic load. We do not develop such a mechanism here, but only mention that it could be driven by user initiated requests for more or less network resources following the opening or closing of new applications (ftp, telnet, netscape *etc.*), with signals from system initiated monitoring of traffics and/or buffers as an alternative or supplement.

In the language of ATM traffic control variations are often said to take place in the cell scale (typically on the order of μs), burst scale (αms), activity scale (αs), session scale (αmin) *etc.*, *e.g.* Bagnoli *et al.* (1994), Hui (1988), Key (1995), Ramamurthy *et al.* (1994), and others. Clearly, buffers are intended only for the cell-, burst- and possibly activity scale, hence modelling of these scales is sufficient from a buffer dimensioning point of view.

2 EXPERIMENTS WITH REAL TRAFFIC AND MODELS

2.1 Background

We defined an experimental test bed based on the following scenario: A user wishes to convey LAN data over an ATM network. The LAN is a 10 Mbps Ethernet, and the user is connected transparently to the ATM network via a 34 Mbps link. Before the LAN packets are delivered over this link to the network, the Ethernet overhead is stripped of, and the remaining data packed into cells. Each 53 octet cell can take 44 octets of Ethernet data, since 4 octets "pay load" are used for AAL3/4 overhead, and the last 5 octets constitute the ATM header.

We implemented a simulator with a server of capacity C and an infinite buffer. Cell arrivals follow sample traces from the Bellcore (1989) material converted to ATM as above, or are drawn from a mathematical model. The traces used were chosen according to length and load: Three time scales were chosen,

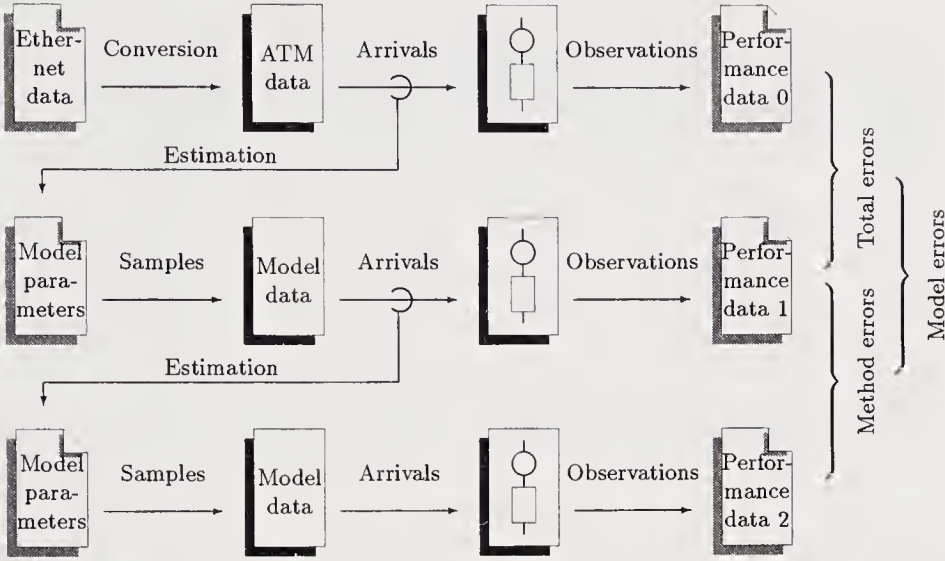


Figure 1 Experiments carried out.

viz. 0.20, 2.00 and 20.0 seconds, and two long term loads, *viz.* 42% and 85% of the actual peak value observed in intervals of those lengths in the entire material made available to us. Finally, for each time scale and load condition were 10 distinct traces selected, resulting in a total of $3 \times 2 \times 10 = 60$ traces. It is observed that the chosen time scales should well cover the normal scope of buffer modelling, *i.e.* cell scale and burst scale variations.

For each trace, the transmission capacity C was set according to the formula for equivalent bandwidth given by Vakili (1993) $C = a(1 - \log a/p)$, where a is the long term average rate (in our case over the entire trace) and p is the peak rate (in our case 34 Mbps). This setting is high enough to ensure that the system is not overloaded, while at the same time it is low enough to let queues build up during the peaks.

2.2 Experiments

In an initial series of runs, each real trace was used as arrival generator in our simulator, figure 1. The trace was run repeatedly in order to emulate a local “steady state”. At each arrival instant we noted the number of cells present in the buffer, which was taken as the sole performance metric for the single server system, denoted in the figure as “Performance data 0” (PD0).

Next, each of the traces were fed into each of the parameter fitting procedures mentioned above. This resulted in one set of model parameters for each trace and each fitting method. The models thus obtained were then used as traffic sources in our simulator, and the performance of the buffer was monitored as before. The observations are shown as “Performance data 1” (PD1) in figure 1.

Noting that the models are fit directly to the traces, one would ideally expect that the models give the same buffer performance as the real traces, *i.e.* PD0 to be equal to PD1. However, it must be remembered

Table 1 Number of infeasible fits.

Model used	0.20 sec.		2.00 sec.		20.0 sec.	
	42% load	85% load	42% load	85% load	42% load	85% load
KMH	—	—	—	—	—	—
TR	—	1	—	—	—	—
MR	6	10	—	5	—	3
HL	4	9	—	4	—	—
RG	3	10	—	8	1	6
DP	5	8	—	1	—	—
BBMP	—	—	—	—	—	—
LL	—	—	—	—	—	—
SDG/1	—	—	—	—	—	—
SDG/4	—	—	—	—	—	—

that the models themselves cannot take all the blame of any differences detected, but some may be due to deficiencies in the parameter estimation *etc.* We may thus say that any difference obtained between a PD0 and PD1 consists of two components: One which is due to the model, and an one which is due to the fitting method and our implementation thereof. We call the former component “model error”, the latter part “method error” and refer to the observed sum as “total errors”.

In order to estimate the two components separately, a new set of experiments was conducted: The above runs for each model and fitting method were monitored and fed to the same fitting procedure as the one used for the model under study, *i.e.*, we fitted each of model to themselves. The resulting set of models were then taken as arrival generators in our simulator, and the buffer performance again monitored as before. The results are indicated as “Performance data 2” (PD2) in figure 1. The fact that the models fitted to are valid by definition in this series of runs means that there are no model errors, but any differences between PD1 and PD2 relate to method errors only. Loosely speaking, we may then obtain the model error by subtracting the method error from the total error.

3 RESULTS

3.1 Validity

For a set of model parameters to be *feasible*, we require that arrival rates are ≥ 0 and transition rates > 0 for MMPPs, and that arrival probabilities are ≥ 0 and ≤ 1 and transition probabilities > 0 and ≤ 1 for MMBPs. The requirements follow from physical interpretations with the added condition that the modulating chain must not be absorbing. Not all fitting methods came up with feasible parameters for all samples. The number of failures are shown in table 1 for each model respectively.

The table shows that these anomalies occur almost solely for direct metrics fitting. The only exception from this rule is one sequence fit, where the modulating chain turned out to be absorbing. It is also

Table 2 Number of abnormal fits.

Model used	0.20 sec.		2.00 sec.		20.0 sec.	
	42% load	85% load	42% load	85% load	42% load	85% load
KMH	1	—	—	—	6	—
TR	—	—	2	4	7	—
MR	—	—	—	—	—	—
HL	—	—	—	—	—	—
RG	—	—	—	—	1	—
DP	—	—	—	—	2	—
BBMP	—	—	—	—	5	2
LL	1	7	—	—	—	—
SDG/1	—	—	—	—	9	—
SDG/4	—	—	5	—	7	1

seen that infeasible parameters are more often obtained at high loads and when fitting to short intervals. Notably, MR and RG failed for all ten traces for the most extreme case in this respect.

Infeasible parameters are explained as follows: Direct fitting methods employ four equations in four metrics from which the four MMPP (MMBP) parameters are found. The output of an MMPP (MMBP) has certain limits regarding the relations between various metrics, and infeasible parameters from a certain trace therefore indicate that the model is incapable of exactly reproducing the metrics of that trace. In this case, one could alternatively find the nearest feasible solution as some kind of best fit. However, our work does not aim at developing or improving fitting methods, but is restricted to testing existing proposals.

Furthermore, for a fitting to be *meaningful*, the resulting traffic model must produce an average queue length that is in the vicinity of the one obtained for the real traffic. We have rather arbitrarily stated that non-meaningful results are those that differ by a factor of 10 or more from the target values. The occurrence of such cases is shown in table 2.

It is seen that abnormal fits almost only occur for sequence fitting and indirect fitting. A closer look at the numbers behind the table reveals mismatches resulting in permanent overloads of the simulated system for the entries referring to sequence fitting. This means that the considered methods, which are iterative, sometimes converge towards a solution that is not correct in terms of average arrival rate. Again, it is beyond the scope of this work to solve the problems behind this phenomenon. For the indirect fitting methods, the abnormal values are less severe, but simply point at weaknesses in the methods as such.

3.2 Accuracy

Metrics

We now remove the infeasible and abnormal fits from our data set and investigate the accuracy of the models with respect to the remaining runs. More precisely, we consider how well the various models and fitting methods can mimic real traffics with respect to dimensioning buffers over the selected time scales.

Let the occupancy of the buffer at an arrival instant be denoted by a stochastic variable Q and define two primary metrics of system performance, *viz.* $E\{Q\}$, the mean buffer occupancy over the entire distribution, and $E\{Q'\}$, the mean occupancy over the tail of the distribution,

$$E\{Q\} = \sum_{k=0}^{\infty} kp(k); \quad E\{Q'\} = \sum_{k=k'}^{\infty} kp'(k)$$

where $p(k)$ refer to the probability of an arriving customer finding k customers already in the queue, and $p'(k)$ is $p(k)$ renormalised over the tail. The tail is defined as all states $k \geq k'$, where k' is the smallest k' such that $\sum_{\kappa=k'}^{\infty} p(\kappa) \leq 10^{-2}$. Note that the latter metric does not refer to a single point, which would have made it very sensitive, but to the *rescaled average* of the last percent of the distribution and thus captures the tail in a wider sense. This number was chosen as a compromise between tail probabilities relevant to buffer dimensioning, typically on the order of 10^{-9} , and simulation feasibility and accuracy.

Adding to the notation, we let Q_i be the performance metric observed from the i th data set in figure 1, *i.e.*, Q_0 refers to the real trace, Q_1 to the fit to the real trace, and Q_2 refers to the fit to the fit. Finally, we define two metrics of the *total error* mentioned in figure 1 as

$$\epsilon_{\text{tot}}(Q) = 1 - E\{Q_1\}/E\{Q_0\}; \quad \epsilon_{\text{tot}}(Q') = 1 - E\{Q'_1\}/E\{Q'_0\}$$

two metrics of the *method error* in the same figure as

$$\epsilon_{\text{met}}(Q) = 1 - E\{Q_2\}/E\{Q_1\}; \quad \epsilon_{\text{met}}(Q') = 1 - E\{Q'_2\}/E\{Q'_1\}$$

and two metrics of the *model error* in the same figure as

$$\epsilon_{\text{mod}}(Q) = E\{Q_2\}/E\{Q_1\} - E\{Q_1\}/E\{Q_0\}; \quad \epsilon_{\text{mod}}(Q') = E\{Q'_2\}/E\{Q'_1\} - E\{Q'_1\}/E\{Q'_0\}$$

Total Errors

Tables 3 and 4 show $\epsilon_{\text{tot}}(Q)$ and $\epsilon_{\text{tot}}(Q')$ respectively for each combination of time scale and load. The numbers shown refer to the average over all valid traces. Rather than providing standard deviations as a supplement, each entry in the table was subject to a *t*-test, *i.e.*, we tested whether the observed average, given the variations between the various traces, could in fact be an observation of a distribution with zero average. The results are depicted in tables 5 and 6 respectively: The number of stars indicate the confidence by which the hypothesis is rejected: three stars mean 99.9% certainty, two stars 99% certainty and one star 95% certainty. No stars thus indicate that the hypothesis cannot be rejected with an error probability below 5%, but *not* that the hypothesis is correct.

It is seen that large errors are frequent, and generally more so for the tail than for the whole distribution. We also note that while many models tend to over estimate the mean of the queue length, they still underestimate the tail, an observation in accordance with observations from heavy tailed traffic.

Table 3 might give the impression that some of the methods based on direct metrics fitting perform reasonably well for short intervals with small, non-significant errors. However, it must be remembered that these values are based on very few actual observations because of the large number of infeasible fits, *cf.* table 1. A similar observation holds for the results of TR in the cases of longer traces.

The same is true for the mean of the tail of the distribution: The only positions with small average errors which pass a test for zero, are those that contain few entries, in particular the case with a time span

Table 3 Total errors observed for the mean of the whole of the distribution.

Model used	0.20 sec.				2.00 sec.				20.0 sec.			
	42% load		85% load		42% load		85% load		42% load		85% load	
KMH	36	**	-136	***	79	***	53	***	88	***	77	***
TR	48	***	-117	**	-6		55	***	2		10	
MR	14		—	—	42	***	34	**	58	***	52	***
HL	23	*	—	—	43	***	41	**	56	***	56	***
RG	11		—	—	43	***	36	*	53	***	53	**
DP	18		-144		75	***	42	***	82	***	67	***
BBMP	52	***	-34	**	82	***	68	***	86	***	78	***
LL	-409	***	-246		-226	**	-232	*	-207	***	-178	***
SDG/1	7		-168	***	65	***	39	***	—	—	68	***
SDG/4	46	***	-70	***	76	***	53	***	83	***	74	***

Table 4 Total errors observed for the mean of the tail of the distribution.

Model used	0.20 sec.				2.00 sec.				20.0 sec.			
	42% load		85% load		42% load		85% load		42% load		85% load	
KMH	-138	**	-700	***	39	**	14		87	***	74	***
TR	-102	**	-648	***	-54		22		3		41	
MR	-204		—	—	-31		3		53	***	48	***
HL	-163	**	—	—	-42		-6		37	**	54	***
RG	-187	**	—	—	-30		7		48	***	64	**
DP	-177	*	-685		32	**	-1		76	***	66	***
BBMP	-54		-336	***	51	***	50	***	80	***	76	***
LL	-1305	***	-1014	*	-640	**	-377	*	-291	***	-205	***
SDG/1	-223	***	-842	***	7		-10		—	—	67	***
SDG/4	-86	*	-532	***	40	**	22	*	81	**	74	***

of 2 seconds and with a long term average load of 85%. An overall conclusion is that the generally large errors make it hard to find a “best model”, and selecting a “worst model” appears equally meaningless.

Method Errors

We will now attempt to get a better idea of the origin of the errors: *i.e.*, if these should be attributed to the models themselves, or if it is just as likely that it is the fitting procedure and our implementation thereof that are to be blamed. Tables 5 and 6 show $\epsilon_{\text{met}}(Q)$ and $\epsilon_{\text{met}}(Q')$ in the same way as above. It is immediately seen that the method errors are by no means small or insignificant for any of the models

Table 5 Method errors for the mean of whole of the distribution.

Model used	0.20 sec.				2.00 sec.				20.0 sec.			
	42% load		85% load		42% load		85% load		42% load		85% load	
KMH	-16	***	-19	***	-14	***	-16	***	-12	***	-15	***
TR	-1	**	-3	***	-3		-2	***	-4	***	-2	**
MR	17		—	—	18		4		28	***	14	*
HL	32	**	—	—	15	**	38	**	7	*	32	***
RG	31	**	—	—	28	*	2		25	**	13	
DP	0		1		1	***	1	*	1	**	1	**
BBMP	-2	*	-1	*	-1		-1	**	-2		-1	***
LL	-9		-46		-16		-40	**	-10		24	**
SDG/1	-26	***	-19	***	-13	*	-23	***	—	—	-28	***
SDG/4	23	***	15	***	31	**	26	***	18	**	33	***

Table 6 Method errors for the mean of tail of the distribution.

Model used	0.20 sec.				2.00 sec.				20.0 sec.			
	42% load		85% load		42% load		85% load		42% load		85% load	
KMH	-15	**	-14	***	-15	*	-11	***	-16	**	-8	***
TR	1		0		0		-1		-2		-5	
MR	14		—	—	15		3		25	***	16	**
HL	29	*	—	—	11	*	35	**	7	*	29	***
RG	30	*	—	—	29	**	4		25	**	10	
DP	22	***	19		24	***	14	***	9	*	3	
BBMP	0		-4		-1		-2		-7		-2	*
LL	-28		-18		-18		-45	**	-7		27	**
SDG/1	-42	***	-21	**	-9		-40	***	—	—	-37	***
SDG/4	28	***	20	***	34	**	24	***	23	**	29	***

but TR and BBMP. If only the mean is considered, the method errors of DP could also be referred to as small.

It is clear that the fitting methods are not particularly stable when it comes to estimating parameters of a model which they essentially have created themselves. This does not mean to say that the formulae provided in the various papers where the methods are put forward are incorrect. What it does say, however, is that the traffic characteristics we are concerned with results in parameters which are hard to estimate. That is, when the trace is fitted to a model for the first time, we get parameters which are in range of hard-to-estimate MMPP-parameters (MMBPs-parameters). The existence of such cases

Table 7 Model errors for the mean of whole of the distribution.

Model used	0.20 sec.				2.00 sec.				20.0 sec.			
	42% load		85% load		42% load		85% load		42% load		85% load	
KMH	52	**	-117	***	93	***	69	***	100	***	92	***
TR	49	***	-115	**	-3		57	***	7		13	
MR	-2		—	—	25		30		31	**	38	***
HL	-9		—	—	28	*	3		49	***	23	**
RG	-21		—	—	15		33		27	**	40	
DP	18		-144		74	***	42	***	81	***	66	***
BBMP	54	***	-34	**	82	***	69	***	89	***	80	***
LL	-401	**	-200		-210	**	-191		-196	**	-202	***
SDG/1	33	*	-150	***	78	***	62	***	—	—	96	***
SDG/4	23		-84	***	45	**	27	**	65	***	41	***

Table 8 Model errors for the mean of tail of the distribution.

Model used	0.20 sec.				2.00 sec.				20.0 sec.			
	42% load		85% load		42% load		85% load		42% load		85% load	
KMH	-123	*	-686	***	54	**	25	**	103	***	82	***
TR	-103	**	-648	***	-54		23		5		46	
MR	-219		—	—	-45		0		28	**	31	**
HL	-192	***	—	—	-53		-41		30	*	25	*
RG	-217	**	—	—	-60	*	3		22	*	55	*
DP	-198	*	-704		8		-15		67	***	63	***
BBMP	-54		-331	***	51	***	52	***	87	***	78	***
LL	-1277	***	-996		-622	**	-332	*	-285	***	-232	***
SDG/1	-181	**	-822	***	15		31	**	—	—	104	***
SDG/4	-114	**	-552	***	6		-2		58	**	44	***

is mentioned already by many of the authors behind the fitting methods, see *e.g.* Meier-Hellstern (1987), Rossiter (1987), and Rydén (1992).

Model Errors

We will finally try to get an idea of the applicability of the models themselves, without respect to the particular fitting method used. Tables 7 and 8 show $\epsilon_{\text{mod}}(Q)$ and $\epsilon_{\text{mod}}(Q')$ in the same way as above.

It is noted that the model errors are of the same order as the total errors and larger than the method errors. Comparing to the former accuracy measures, the differences between the various fitting methods

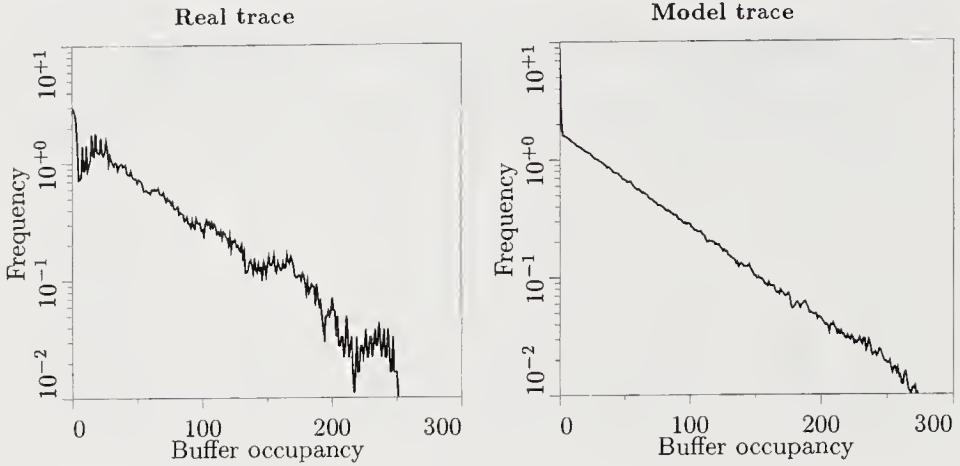


Figure 2 Comparison of buffer occupancies resulting from real and artificial traces for a trace of 2.00 seconds with 85% load. The artificial trace is produced by an SPP fitted by means of the HL-method. The upper plots refer to the whole trace and the lower ones to the first 2% of the trace.

remain, hence our attempt to separate the fitting method from the model is not entirely successful. The tables clearly show that no model succeeds in accurately predicting both the mean of the whole distribution and of its tail. As before, low values are almost exclusively noted in conjunction with a large number of failed fits. This makes it hard to point at any particularly successful or promising model.

Some Detailed Results

To get a deeper understanding of the results, we have arbitrarily selected a case for which reasonably good agreement was obtained in the study above, *viz.* direct fitting for 2 second intervals.

Two two plots in figure 2 show the buffer occupancy distributions for the real and artificial traffics respectively. The two curves clearly appear quite similar at a first glance. On the other hand, at a closer look, the two differ around zero and in their tails: The real data has a lower value at the origin and exhibits a knee at the tail, while the model data has a higher value in the origin and the tail is straight. These findings are in agreement with what has been suggested by Pruthi (1995) and others: Markov-type models result in buffer occupancy distributions with exponential tails, while many real traffics result in power-law tails.

Looking at the similarity of the curves, these differences might be regarded as minor details, but it must be remembered that the models are to be used for determining loss probabilities on the order of 10^{-9} . Using our performance metrics, the similar shapes are reflected by the means, $E\{Q_0\} = 53.46$ for the real data and $E\{Q_1\} = 49.39$ for the model, while the different tails results in $E\{Q'_0\} = 251.8$ and $E\{Q'_1\} = 462.2$ respectively.

Figure 3 shows plots of the number of arrivals within intervals, $N(t, t + \Delta t)$ *vs.* t for the real and

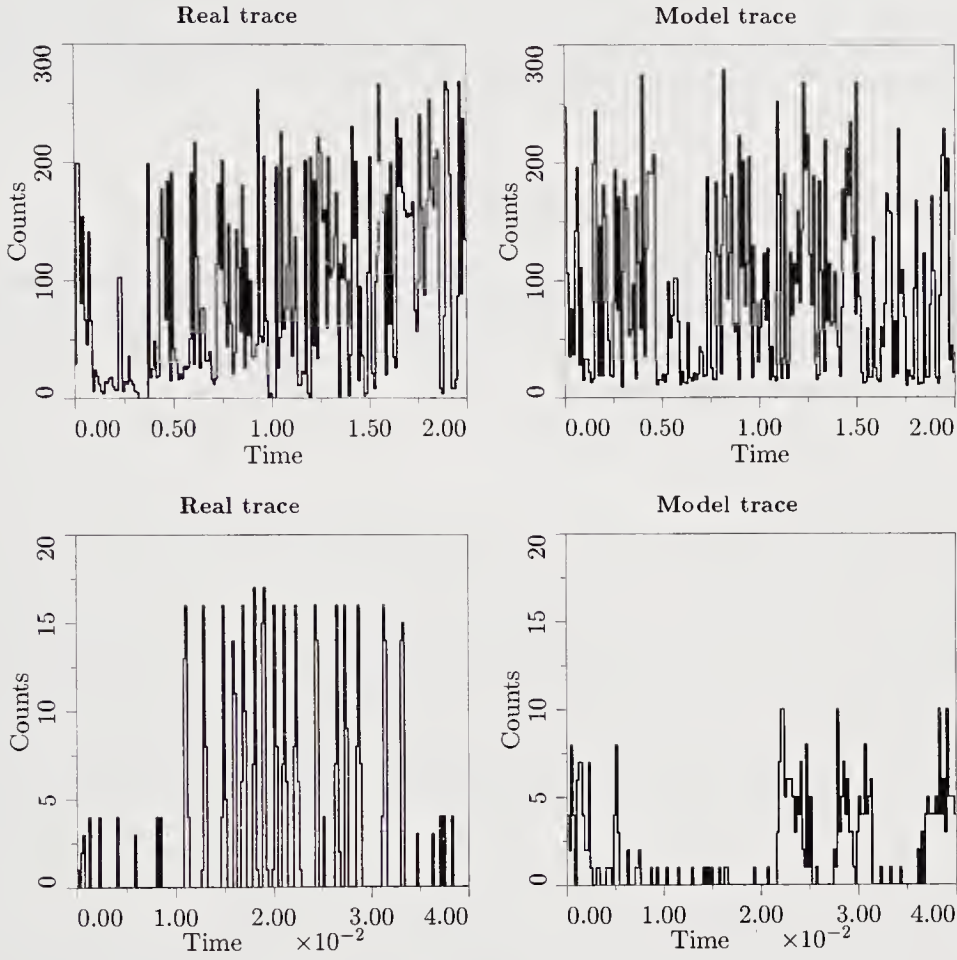


Figure 3 Comparison of real and artificial traces for a trace of 2.00 seconds with 85% load. The artificial trace is produced by an SPP fitted by means of the HL-method. The upper plots refer to the whole trace and the lower ones to the first 2% of the trace.

artificial traces. The upper plots refer to the whole trace and the lower ones show the first 2% of the trace in more detail.

It is noted that there are no apparent, fundamental differences in the large time scale between the real trace and the artificial one. However, it is also seen from the two lower plots that this statement does not seem to hold in the higher frequencies. This again confirms the results by Pruthi (1995) and others, that

the lower frequencies rule the average queue length, and hence the average delay, while higher frequencies are critical to the tail of the queue and therefore to the loss probability in case of an infinite buffer.

Noting a reasonable agreement for the overall mean, but a less good for the tail, it is tempting to conclude the models covered by our investigation might be more useful for calculating delays than losses. Tables 3-8 do, however, not support such a conclusion in general.

4 CONCLUSIONS AND FURTHER WORK

We have tried a number of methods proposed for fitting an MMPP (MMBP) to observed traffic data. Sixty data sets were extracted from the Bellcore Ethernet measurements according to length and local average load, so that short, medium and long periods of both light and heavy loads were tried. We then compared the performance of the buffer of a single server system when subject to the real traffic and when subject to traffic from the fitted model.

Several cases of infeasible parameters were recorded. A simple solution to this problem might be to restate the various methods as constrained optimisation problems, where a best fit under the condition of feasible parameters is determined.

It was found that the two cases differ significantly in terms of buffer occupancy, and that these differences are caused by deficiencies in the different fitting methods and possibly also by limitations in the models themselves.

Restricting ourselves to shorter time spans of up to two seconds, it was noted that direct metric fitting methods produced the smallest errors and resulted in traces that appeared identical to the real ones on a large time scale. It is therefore concluded that the most promising candidates for a "good model and fitting method" are found in the group, though further work is needed to clarify the importance of and methods for fitting a wider range of frequencies before "safe" models can be devised.

Moreover, if time scales of two seconds can be models, there is no reason why shorter time spans could not be mastered too if the problem of fitting to a small data set can be solved. On the other hand, it also seems clear from the tables that the chances of finding small MMPPs (MMBPs) that remain valid for time scales of 20 seconds and above are fairly slim.

This work is different from what is normally published on modelling and fitting bursty traffics in that we use *real* traffic. This fact means that we have had to develop new practices and faced difficulties in ending up with neat conclusions regarding a perfect model and fitting method.

We believe, however, that there is enough real data available to stop validating models against models, but actually use real data instead. This work constitutes a first step in this direction, and we hope to have inspired others than ourselves to continue this important work. We can identify a large number of issues that need to be looked into, for instance

- Finding traffic characteristics which are relevant from the point of view of buffer dimensioning and for which simple and robust estimation techniques can be devised. Some theoretical proposals are given in *e.g.* Andrade *et al.* (1991) and Grünenfelder *et al.* (1994).
- Finding fitting methods for these characteristics which always come up with the best physically feasible fit. A first approach is to somewhat modify the methods tried here.
- A repeat of our investigation but with much more than ten samples per time scale and utilisation level.
- Repeating our investigation as above for other traffic sources than the Bellcore Ethernet.

We also note that if modelling short time scale variations shall be useful, a number of issues must be resolved, for example how to handle the long term variations in practice. Possible candidates includes reallocations of transmission capacity according to network predictions (*e.g.* by monitoring cell flow or buffer contents) or users' requests (*e.g.* when opening or closing particular applications or application modes).

5 ACKNOWLEDGEMENT

Thanks to Dr. Parag Pruthi at the Royal Institute of Technology (Sweden) for providing the extracts from the Bellcore measurements, to Dr. Tobias Rydén at the Lund Institute of Technology (Sweden) for providing the code for the TR method, and to Dr. Claes Jøgréus at the University of Karlskrona/Ronneby (Sweden) for discussions on statistical testing.

REFERENCES

- Andrade, J., Burakowski, W., and Villen-Altamirano, M. (1991) Characterization of Cell Traffic Generated by an ATM Source, in *Teletraffic and Datatraffic in a Period of Change*, Elsevier.
- Arvidsson, Å., Berry, L. and Harris, R. (1991) Performance Comparison of Bursty Traffic Models, in *Proc. Austr. Broadb. Switching and Services Symp.*, Sydney.
- Andersen, A., Jensen, A. and Friis Nielsen, B. (1995) Modelling and Performance Study of Packet-Traffic with Self-Similar Characteristics over Several Time Scales with Markovian Arrival Processes, in *Proc. 12th Nordic Teletraffic Sem.*, Helsinki.
- Bagnoli, G., Listanti, M. and Winkler, R. (1994) Cell Level and Frame Level Performance of Traffic Control Schemes for No Resource Reservation Data Communications in ATM Networks, in *The Fundamental Role of Teletraffic in the Evolution of Telecommun. Netw.*, Elsevier.
- Bellcore, Measurements made on August, 29 1989 at 11.25 a.m. at *Bellcore Research and Engineering Centre*, Morristown.
- Bonomi, F., Meyer, J., Montagna, S. and Paglino, R. (1994) Minimal On/Off Source Models for ATM Traffic, in *The Fundamental Role of Teletraffic in the Evolution of Telecommun. Netw.*, Elsevier.
- Frater, M., Tan, P. and Arnold, J. (1994) Variable Bit Rate Video Traffic on the Broadband ISDN: Modelling and Verification, in *The Fundamental Role of Teletraffic in the Evolution of Telecommun. Netw.*, Elsevier.
- Grünenfelder, R. and Robert, S. (1994) Which Arrival Law Parameters are Decisive for Queueing System Performance, in *The Fundamental Role of Teletraffic in the Evolution of Telecommun. Netw.*, Elsevier.
- Gusella, R. (1991) Characterizing the Variability of Arrival Processes with Indexes of Dispersion. *IEEE J. Sel. Areas in Commun.*, 9, 203–211.
- Heffes, H. and Lucantoni, D. (1986) A Markov Modulated Characterization of Packetized Voice and Data Traffic and Related Statistical Multiplexer Performance. *IEEE J. Sel. Areas in Commun.*, 4, 856–8.
- Hui, J. (1988) Resource Allocation for Broadband Networks. *IEEE J. Sel. Areas in Commun.*, 6, 1598–608.
- Jain, R. and Routhier, S. (1986) Packet Trains — Measurements and a New Model for Computer Network Traffic. *IEEE J. Sel. Areas in Commun.*, 4, 986–95.

- Key, P. (1995) Modelling, Measurement, and Connection Admission Control in ATM Networks, in *Proc. 9th ITC Specialists Seminar on Teletraffic Modelling and Measurement in Broadband and Mobile Communications*, Leidschendam.
- Lee, J. and Lee, B. (1992) Performance Analysis of ATM Cell Multiplexer with MMPP Input. *IEICE Trans. on Commun.*, **E75-B**, 709–14.
- Leland, W., Taquq, M., Willinger, W. and Wilson, D. (1994) On the Self-Similar Nature of Ethernet Traffic (Extended Version). *IEEE/ACM Trans. on Networking.*, **2**, 1–15.
- Meier-Hellstern, K. (1987) A Fitting Algorithm for Markov-Modulated Poisson Processes Having Two Arrival Rates. *Eur. J. Op. Res.*, **29**, 370–7.
- Park, D. and Perros, H. (1994) m-MMBP Characterization of the Departure Process of an m-MMBP/Geo/1/K Queue, in *The Fundamental Role of Teletraffic in the Evolution of Telecommun. Netw.*, Elsevier.
- Paxson, V. and Floyd, S. (1994) Wide-Area Traffic The Failure of Poisson Modeling, in *Proc. ACM Sigcomm 94*, London.
- Pruthi, P. (1995) An Application of Chaotic Maps to Packet Traffic Modeling. *Ph.D. dissertation*, Dept. of Teleinformatics, Royal Inst. of Tech., Stockholm.
- Ramamurthy, G. and Dighe, R. (1994) Analysis of Multilevel Hierarchical congestion Controls in B-ISDN, in *The Fundamental Role of Teletraffic in the Evolution of Telecommun. Netw.*, Elsevier.
- Rose, O. and Frater, M. (1994) A Comparison of Models for VBR Video Traffic Sources in B-ISDN, in *Proc. IFIP TC6/WG6.2 Broadband Commun. '94*, Paris.
- Rossiter, M. (1987) A Switched Poisson Model for Data Traffic. *Austr. Telecommun. Res.*, **21**, 53–7.
- Ryden, T. (1992) Parameter Estimation for Markov Modulated Poisson Processes. *Technical report (TFMS-LUTNFD23083)*, Department of Mathematical Statistics, Lund Institute of Technology, Lund.
- Solé, J., Domingo, J. and Garcia, J. (1990) Modelling the Bursty Characteristics of ATM Cell Streams, in *Proc. IEE Int. Conf. on Integr. Broadb. Services and Netw.*, London.
- Vakil, F. (1993) A Capacity Allocation Rule for ATM Networks, in *Proc. IEEE Globecom '93*, Houston.

PART TWO

Traffic and Congestion Control

A Congestion Control Mechanism for Connectionless Services offered by ATM Networks

S. Halberstadt[†], D. Kofman[†] and A. Gravey[‡]

*[†]Ecole Nationale Supérieure des Télécommunications
Networks Department.*

46 rue Barrault, 75634 Paris Cedex 13, France.

Telephone: 33-1-45-81-75-70, Fax: 33-1-45-81-75-76.

email: halbers,kofman@res.enst.fr

*[‡]France Télécom-Centre Nationale d'Etudes des Télécommunications
CNET/LAA/EIA/EVP, Route de Trégastel, 22301 Lannion Cedex,
France.*

Telephone: 33-96-05-39-84, Fax: 33-96-05-39-45.

email: graveya@lannion.cnet.fr

Abstract

We propose a traffic management mechanism for connectionless networks on top of ATM infrastructures. The mechanism combines flow control at the packet layer (connectionless layer) and dynamic bandwidth allocation of the ATM connections interconnecting the connectionless servers of the connectionless network. Optimal mechanisms are obtained through Markov decision processes for a model of two tandem queues. The obtained bandwidth gain motivates the analysis of such mechanisms in a more realistic model. The simulation of a more detailed model of a connectionless network allows us to conclude on the favorable impact of dynamic resource allocation on the bandwidth gain and on the reduction of the sensitivity of the performances of the network with respect to the characteristics of the traffic. The traffic management mechanism implemented in the simulator are motivated by the optimal mechanism obtained using the analytical model.

Keywords

ATM Networks, connectionless services, traffic management, Markov decision processes

1 INTRODUCTION

One of the first expected applications of ATM networks is LAN interconnection. Since ATM is a connection-oriented transfer mode, the provision of connectionless services (LANs are connectionless) is a challenge for ATM networks. ITU recommendation I211 has proposed two approaches to offer a connectionless service in B-ISDN, namely the direct and indirect methods. This paper deals with the first one, which consists of introducing ConnectionLess Servers (CLSs) into B-ISDN. These CLSs are interconnected through ATM connections thus forming an "overlay network". The functions of the CLSs are mainly to route packets (datagrams) and to manage connectionless traffic (Vickers, 1994).

Congestion control in this network can be achieved as in classical datagram networks by means of mechanisms combining dynamic routing and flow control. When ATM connections are used instead of leased lines, a third traffic management approach can be introduced : the dynamic bandwidth allocation. By taking advantage of the flexibility of ATM connections, it may allow a gain in bandwidth utilization.

Mechanisms using dynamic bandwidth allocation have been proposed in previous papers (Gallassi, 1992) (Ileijenk, 1992)(Mongiovi, 1991)), but mainly for the indirect approach (no CLS). In (Gallassi, 1992), it is argued that the proposed mechanism works well in the direct case, but no study of performance is given. In (Yamamoto, 1993), a mechanism combining flow control and dynamic bandwidth allocation is proposed for both direct and indirect approaches; however, in the direct case, Yamamoto and al. propose to use dynamic bandwidth allocation only on the access links. They argue that only a simple feedback type flow control mechanism is necessary between CLSs, because fluctuation of traffic on such links is small due to the high degree of multiplexing that is achieved. Unfortunately, as shown in (Van den Berg, 1995), the high unpredictability of connectionless traffic makes the effective dimensioning extremely difficult in case of fixed bandwidth links.

The goal of this paper is, on one hand, to show that a non negligible gain in the utilization of bandwidth can be obtained by using mechanisms combining flow control and dynamic bandwidth allocation and, on the other hand, to determine the mechanisms allowing such a gain. The gain is studied with respect to the situations where no mechanism or only flow control is used.

The structure of this paper is as follows. In Section 2, a simple model of two tandem queues is studied in order to gain insight into the type of mechanisms that optimize the bandwidth gain. A Markov decision process approach is used to model the system. In Section 3, a more realistic network is studied by simulation and it is shown that the proposed mechanism, motivated by the results in Section 2, allows a gain in bandwidth utilization while assuring some QoS constraints, and reduces the sensitivity of the performances of the network with respect to the traffic characteristics. We conclude in Section 4.

2 TANDEM QUEUES

2.1 The queueing system

The system we study consists of two tandem queues with finite buffers (see Figure 1). Queue 1 may represent an individual source or a ConnectionLess Server whereas queue 2, which is the queue of interest, represents a CLS. Packets (datagrams) enter the system following an Interrupted Poisson Process (IPP). The choice of an IPP was made in order to capture the bursty nature of connectionless traffic. A more realistic arrival process would be of little interest for this study, which aims to obtain qualitative results on the gain that can be achieved by using dynamic bandwidth allocation when traffic may be bursty. Service times are exponentially distributed. The service rates vary dynamically, representing, respectively the flow control for the first queue and the dynamic bandwidth allocation for the second. More precisely, a controller is placed at queue 2 and may decide either to control queue 1 from μ_2 (normal rate) to μ_1 (controlled rate, with $\mu_1 < \mu_2$), to ask for an increase of service rate for queue 2 from ν_1 (normal rate) to ν_N (maximum rate), or to leave the service rates at their normal values (μ_2 and ν_1).

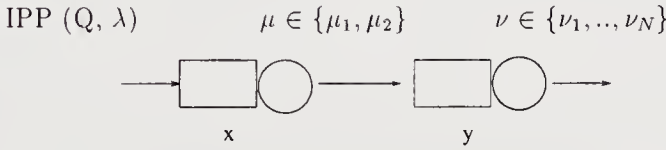


Figure 1 Tandem controlled queues.

The notations are the following:

- $Q = \begin{pmatrix} -q & q \\ r & -r \end{pmatrix}$ is the infinitesimal generator of the phase of the IPP.

This means that q (respectively r) is the rate of passage from silent period to active period (respectively from active period to silent period) of the arrival process.

- λ is the intensity of the arrival process when active (i.e. the peak rate of the arrival process).

From the preceding definitions, we observe that the mean rate of the IPP is equal to

$$\frac{\frac{1}{r}}{\frac{1}{r} + \frac{1}{q}} \lambda = \frac{q}{q + r} \lambda.$$

- b is the activity parameter of the arrival process, which we define as the ratio mean rate/ peak rate; it is given by $b = \frac{q}{q + r}$.
- μ_i , $i \in \{1, 2\}$ are the possible service rates of queue 1.
- N is the number of allocation levels for the server of queue 2 (i.e N is the number of possible service rates of queue 2).
- ν_j , $j \in \{1, \dots, N\}$ are the possible service rates of queue 2.
- K_1 (respectively K_2) is the capacity of queue 1 (respectively queue 2).

We denote by (x, y, z) a state where x is the number of packets in queue 1, y is the number of packets in queue 2 and z the phase of the IPP (0 if it is silent, 1 if it is active).

Our goal is to find an *efficient* and *simple* mechanism combining flow control and bandwidth allocation. We therefore use the theory of Constrained Markov Decision Processes (Puterman, 1994) in order to determine an optimal mechanism satisfying some QoS constraints. The optimality is defined with respect to a cost function, which is an increasing function of the service rate of queue 2.

In the following subsection, the studied controlled process is described.

2.2 The Markov decision process

Considered is a continuous time Markov decision process with the following characteristics:

- the *static space* is $S = [0, K_1] \times [0, K_2] \times \{0, 1\}$.
- the *action space* is $A = \{(\mu_i, \nu_j), i \in \{1, 2\}, j \in \{1, \dots, N\}\}$.
- the *transition rates* are the same as for a model with fixed rates, with μ and ν depending on the chosen action. More precisely, the evolution of the process is determined by a family of infinitesimal generators indexed by the action space.
- the *instantaneous cost function*, denoted by $c[(x, y, z); \mu, \nu]$, which is given below.

The evolution of the process is as follows. At each transition time t of the process, a decision is taken, that is a pair of service rates is chosen in A , say (μ_t, ν_t) . The process then evolves from current state (X_t, Y_t, Z_t) in S to its next state $(X_{t'}, Y_{t'}, Z_{t'})$ with rates given by the infinitesimal generator $Q(\mu_t, \nu_t)$. Between t and t' , a cost is incurred at rate $c(X_t, Y_t, Z_t; \mu_t, \nu_t)$. Once the transition into the next state has occurred, a new decision is taken.

The objective is to find a policy (i.e. a sequence of actions) π minimizing the average cost, i.e. the following function (Puterman, 1994):

$$V_\pi(x, y, z) = \limsup_{T \rightarrow \infty} \frac{1}{T} E_\pi^{(x, y, z)} \left\{ \int_{t=0}^T c(X_t, Y_t, Z_t; \mu_t, \nu_t) dt \right\},$$

under the constraint:

$$\limsup_{T \rightarrow \infty} \frac{1}{T} E_\pi^{(x, y, z)} \left\{ \int_{t=0}^T (1_{\{X_t=K_1\}} + 1_{\{Y_t=K_2\}}) dt \right\} \leq \alpha.$$

Here 1_X is the indicator function of the random variable X and $E_\pi^{(x, y, z)}$ means the expectation with respect to the probability induced by policy π and initial state (x, y, z) (see (Puterman, 1994), chapter 2, for a rigorous presentation of the probabilistic framework for Markov decision processes). The considered constraint imposes that the probability of saturation (i.e the probability that one of the buffers is full) does not exceed α . Under irreducibility assumptions one knows (Puterman, 1994) that the optimal average cost does not depend on the initial conditions; furthermore, there exist several algorithms allowing to compute the optimal cost and policies.

The computation of an optimal policy requires first to transform the initial problem into a discrete-time one; to that purpose We use the uniformization technique (Serfozo, 1979) to transform the initial problem into a discrete-time one. This approach, which is

very common in Markov decision process theory, allows to obtain a discrete time Markov decision process which is equivalent to the initial one in the sense that every optimal policy for one problem is optimal for the other.

We then use the classical formulation of Markov decision processes using Linear Programming (Derman, 1970). This formulation, which allows us to easily take into consideration the constraint on the probability of saturation, leads to the following linear programming problem:

$$\text{minimize } \sum_{i=1}^2 \sum_{j=1}^N \sum_{(x,y,z) \in S} c[(x,y,z); \mu_i, \nu_j] \xi[(x,y,z); \mu_i, \nu_j].$$

subject to the following constraints:

$$\xi[(x,y,z); \mu_i, \nu_j] \geq 0, \quad \forall (x,y,z) \in S, \quad \forall i = 1, 2 \quad \forall j = 1, \dots, N,$$

$$\sum_{i=1}^2 \sum_{j=1}^N \sum_{(x,y,z) \in S} \xi[(x,y,z); \mu_i, \nu_j] = 1,$$

$$\sum_{i=1}^2 \sum_{j=1}^N \xi[(x,y,z); \mu_i, \nu_j] = \sum_{i=1}^2 \sum_{j=1}^N \xi[(x,y,z'); \mu_i, \nu_j] \sum_{(x',y',z') \in S} P[(x,y,z')|(x,y,z); \mu_i, \nu_j],$$

$$\sum_{i=1}^2 \sum_{j=1}^N \sum_{(x,y,z) \in S} [1_{\{x=K_1\}} + 1_{\{y=K_2\}}] \xi[(x,y,z); \mu_i, \nu_j] \leq \alpha,$$

where P is the probability matrix obtained after uniformizing the process.

In the following subsection, the obtained results are presented.

2.3 Results

The objective of this study is to examine the impact of dynamically allocating bandwidth. To that purpose, a first system (system **FC** - Flow Control), where only one service rate is available for queue 2 (i.e. $N=1$) is compared with a second one (system **FCDA** - Flow Control and Dynamic Allocation), where the service rate of queue 2 may take several values ($N > 1$).

For system FC, we look for the minimum service rate ν for which the constraint can be satisfied (this means, for which there exists a policy such that the constraint is satisfied). Linear programming informs indeed about the feasibility of the constraint.

For system FCDA, the service rate of queue 2 is allowed to vary in a wide range (see next paragraph) so that feasibility is always achieved.

Choice of parameters:

Here are the values chosen for the parameters of the model. Although they do not reflect realistic figures, mainly because of the size of the state space when considering large buffers, reasonable proportions were kept between the different values. The qualitative results that we obtained in all the experiments we carried out were quite similar.

- The capacities of queue 1 and 2 are $K_1 = 15$ and $K_2 = 25$.

- The following values are considered for the possible service rates of queue 1 and queue 2:
 $\mu_1 = 5, \mu_2 = 200$.
 $N = 5$ and $\nu_1 = 5, \nu_2 = 50, \nu_3 = 100, \nu_4 = 150, \nu_5 = 200$.
- The characteristics of the arrival process vary throughout our study as follows: arrival processes with different activity parameter (from 0.025 to 0.5) and mean burst duration (5 and 10 packets) values are considered.
 The whole set of parameters of the IPP is set by imposing a constant mean rate equal to 5.
- the target saturation probability α is equal to 10^{-3} .
- The considered instantaneous cost function is $c[(x, y, z); \mu, \nu] = \nu^2$. The choice of a function of ν increasing faster than linearly is made in order to reflect the following fact. When dealing with dynamically allocated bandwidth for bursty traffic, a request for a large amount of bandwidth costs more to the network operator than several requests for smaller amounts. This is because the multiplexing gain that can be achieved strongly decreases with the peak rate of the considered bursty traffic (Roberts, 1991).
 The square root of the average cost, which is denoted by "normalized cost", is taken as the measure of the required bandwidth.

Influence of the traffic characteristics on optimal cost

Figure 2 shows the normalized cost obtained for the two mechanisms as a function of the activity parameter of the arrival process. The left-hand figure is obtained for bursts with mean length 5 packets and the right-hand one for bursts with mean length 10 packets.

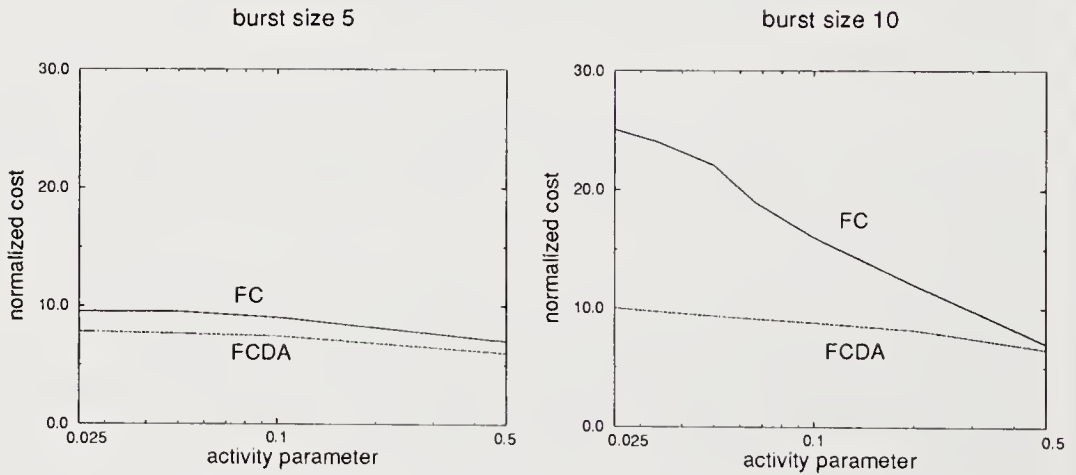


Figure 2 Comparison of FC and FCDA mechanisms.

It can be observed that, in both cases, there is a gain in cost (that is in utilization of bandwidth) when dynamic bandwidth allocation is used. This gain is an increasing function of the activity parameter (for bursts of size 5, this gain ranges from 14.5% for activity parameter 0.5 to 17.8% for activity parameter 0.025; for bursts of size 10, it ranges from 7% for activity parameter 0.5 to 60% for activity parameter 0.025). The gain

in bandwidth utilization appears to be considerable if the activity parameter is low (i.e. peak rate is much larger than mean rate) on the considered links. This allows to hope for a substantial gain in the utilization of bandwidth in CLS network where it is not possible to assert that activity parameters will be high.

It can also be concluded that the obtained mechanism highly reduces the sensitivity of the performances of the network with respect to the traffic characteristics. This simplifies the dimensioning of the resources of the network.

Structure of optimal policies

From the preceding paragraph, it can be concluded that dynamic bandwidth allocation seems to lead to a substantial gain in bandwidth utilization. In this paragraph, we aim to gain insight about the design of efficient mechanisms. To that purpose, the structure of the optimal policies obtained using linear programming is studied. Figures 3 and 4 represent the optimal actions as a function of the occupancies of buffer 1 and 2, when the arrival process is active ($z=1$). The burst size is equal to 8 packets and the activity parameter varies from 0.1 to 0.025.

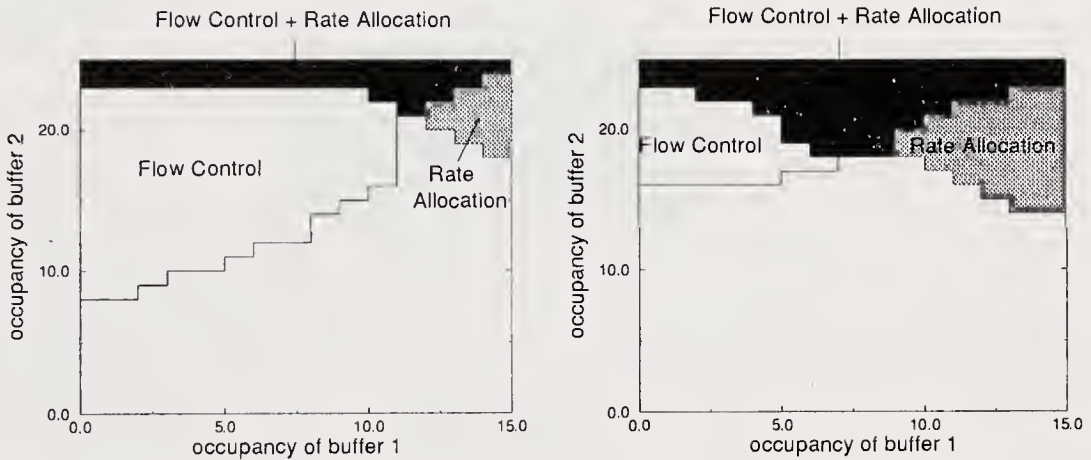


Figure 3 Optimal FCDA mechanism (left: activity parameter 0.1, right: activity parameter 0.05).

It has to be noticed that, the larger the peak rate of arrival process compared to the mean rate, the more the obtained optimal mechanism relies on rate allocation and the less on flow control.

The considerable gain that can be achieved using the optimal policy incites to the analysis of a more realistic model of a CLS network using flow control and dynamic bandwidth allocation. We focus on the case of low activity parameter and so we analyze a traffic management mechanism motivated by the right component of Figure 4. The optimal mechanism represented in this figure is approximated by a simpler one of the type represented in Figure 5.

The proposed traffic management mechanism is described in detail in the following section.

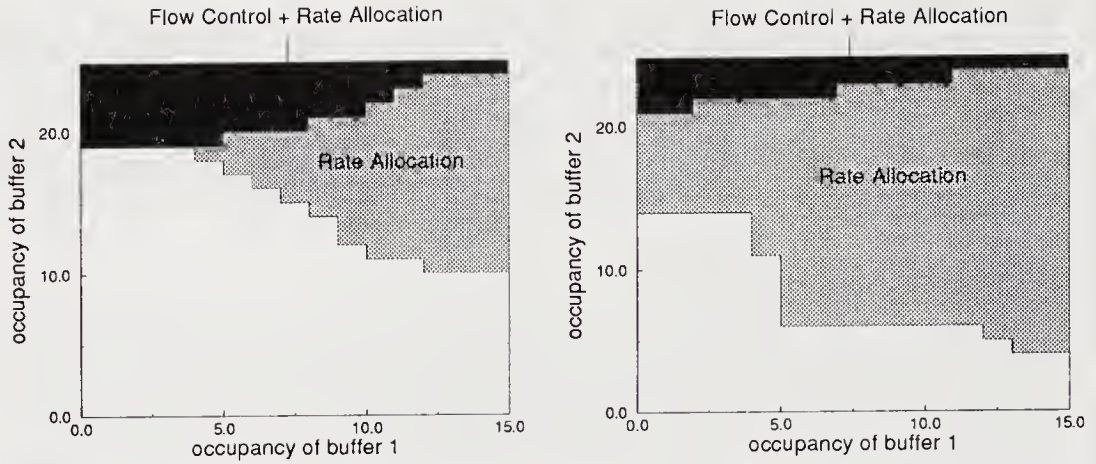


Figure 4 Optimal FCDA mechanism (left: activity parameter 0.033, right: activity parameter 0.025).

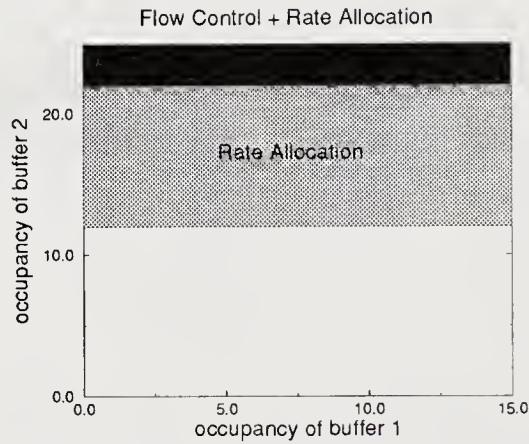


Figure 5 Approximated FCDA mechanism.

3 SIMULATION STUDY

3.1 The simulation model

The considered model consists of a network of CLSs fed by bursty sources (see Figure 6). Packets of constant size (equal to 1.5 Kbyte) arrive, following an Interrupted Poisson Process, to the buffer of a source before being transmitted to the network. A very simple topology is considered for the CLS network : two CLSs (CLSs B and C) transmit packets to a third one (CLS A). There is a non negligible propagation time between sources and CLSs and between CLSs. This propagation time also affects information corresponding to the flow control and bandwidth allocation procedures. The capacity of all CLSs buffers is equal to 1000 packets (1.5 Mbytes). These buffers are supposed to be dedicated to an

output link of the corresponding CLS. Different cases will be analyzed for the sources buffer size, as described later.

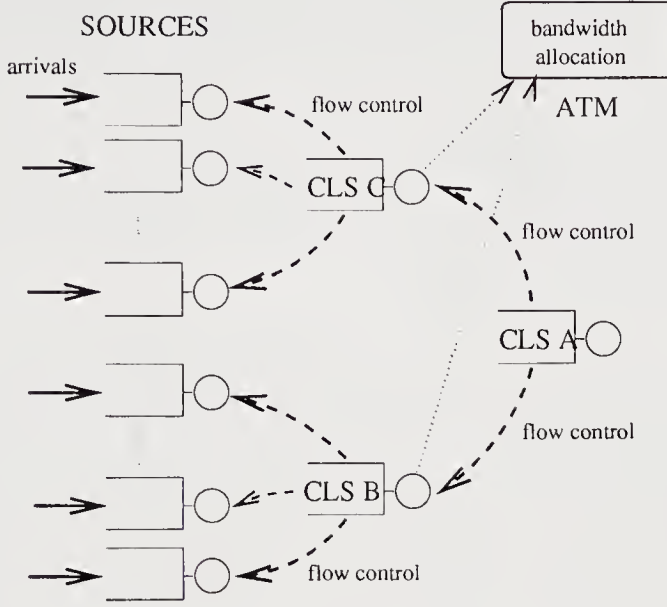


Figure 6 The simulation model.

The proposed **Flow Control** mechanism is the following. When the occupancy of the buffer of a CLS reaches the level T_{ctrl} , this CLS asks for a reduction of the transmission rates on its input links. Each CLS measures the arrival rate in its buffer and reduces its input rate so as to obtain a fixed ratio controlled input rate/output rate (which is denoted by ρ). This control is exercised until the occupancy of the buffer of the congested CLS passes below the level T_{rlse} ($T_{rlse} < T_{ctrl}$).

The proposed **Dynamic Allocation** mechanism is the following. When the occupancy of the buffer of a CLS reaches the level $S_{alloc} < T_{ctrl}$, this CLS asks for an increase of the rate on its output link. Depending on the situation of ATM links, this requests can be accepted or not. Each time it is accepted, the rate of the output link is increased by a fix amount P Mbit/s. The congested CLS waits some time F (for filter) before requesting again for more bandwidth if still congested. Congestion is considered to be over as soon as the occupancy of the CLSs buffer passes below the level S_{rlse} ($S_{rlse} < S_{alloc}$). In this case, the rate of the output link takes its initial value (which corresponds to the minimum of the possibles values of this rate). The probability of refusals ($P_{refusal}$) is an increasing function of the already allocated bandwidth. More precisely, it is supposed to be of the following form: $P_{refusal} = \min(((\nu_r - \nu_i)/P)P_f, 1)$, where ν_i is the initial rate and ν_r is the requested rate.

If an increase is refused, the CLS waits some time W before requesting again for bandwidth if still congested.

Choice of parameters

Here is the list of the chosen parameters for the simulation study. Since some parameters, especially concerning the arrival processes into the sources, are going to change during our experiments, the given values are valid unless otherwise stated.

- number of sources : 80
- mean rate of sources : 1 Mbit/s
- peak rate of sources : 100 Mbit/s
- mean burst duration : 200 ms
- $P_f = 0.1$, $P = 30$ Mbit/s, $W = 70$ ms, $F = 20$ ms, $\rho = 0.8$.
- $S_{\text{alloc}} = 600$ packets, $S_{\text{rlse}} = 500$ packets
- $T_{\text{ctrl}} = 900$ packets, $T_{\text{rlse}} = 850$ packets

A few remarks have to be made about the choice of these thresholds. The dimensioning of T_{ctrl} is mainly linked to the distance between sources and CLSs or between CLSs, i.e. to the time between congestion notification and effective decreasing of the rate. In principle, one could choose this threshold so that, even if all input links of the congested CLS are “active”, i.e. emitting packets, all packets emitted before the reaction of the sources are absorbed and so there are no lost packets. This would require too large capacities for the buffers of the CLSs or a poor utilization of these buffers most of the time. The choice of 900 corresponds to a relatively reasonable compromise between utilization of the buffer and efficiency of the control mechanism.

There is an alternative in the choice of T_{rlse} : if it is too close from T_{ctrl} , this will lead to numerous oscillations between control and release of the control* ; if it is too far, this will impose very long periods of control to the controlled sources or CLSs.

The choice of S_{alloc} and S_{rlse} conditions the frequency of allocation requests, so that the alternative is between efficiency and frequency of allocations. S_{rlse} is not to be chosen too low for another reason, which is the utilization of the allocated bandwidth: if S_{rlse} is too low, there is a risk that the buffer will empty between the instant when the decision of decreasing the output rate is taken and the instant when this rate is actually decreased. In this case, the utilization of the supplementary allocated bandwidth will not be maximized.

3.2 Simulation results

In this section it is shown, by means of numerical results, that the global traffic management mechanism we propose allows a gain in the utilization of bandwidth.

The simulations were realized with Simscript II.5.

Influence of the characteristics of input traffic

We first compare the system without any control (curves denoted NM, for No Mechanism), with traffic control (curves denoted FC, for Flow Control) and with traffic control and dynamic bandwidth allocation (curves denoted FCDA, for Flow Control and Dynamic Allocation). The systems without any control, with control and with control and dynamic

*This generates a non-negligible traffic of control messages.

bandwidth allocation will also be denoted, respectively, by NM, FC and FCDA. In the first experiments, the buffers of the sources are supposed to have an infinite capacity.

In all the cases the “required bandwidth” is defined as the minimum bandwidth necessary to guarantee a quality of service defined by a loss probability of 10^{-4} and a maximum delay less than 30ms for 95% of the packets (here the delay is the time a packet spends in the system from the instant it enters a source to the instant it leaves CLS A).

Figure 7 shows the bandwidth required for the output links of CLSs B and C as a function of the mean burst duration (the mean silence duration is varied in the same proportion as the mean burst duration in order to maintain the mean arrival rate). For system FCDA, this bandwidth is by definition the weighted average of the different rates which are used, with weights being given by the proportion of time each rate is used. The main conclusions are the following:

- the FCDA mechanism allows a saving which, for mean burst duration of 400ms, is of 39% compared with system NM and of 9% compared with system FC.
- the gain of the FCDA mechanism with respect to FC mechanism is an increasing function of the mean burst duration. Since the maximum burst duration considered corresponds to a burst length of 4.8 Mbytes, some greater values of this parameter, which are not unrealistic in the context of LAN interconnection, could lead to a more substantial gain.
- The slope of the curve corresponding to the FCDA mechanism is lower than for the other two mechanisms, which reflects a lower sensitivity of this mechanism to burst length, which is a difficult parameter to predict.

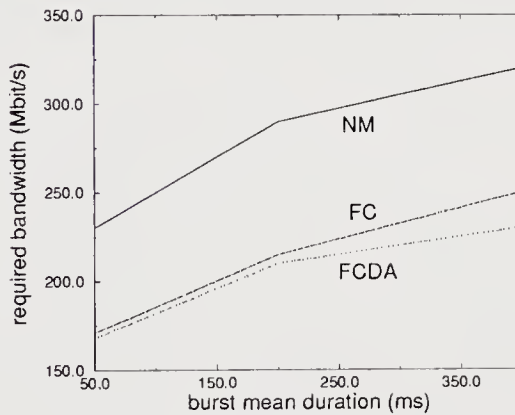


Figure 7 Influence of mean burst duration.

Figure 8 shows the influence of the activity parameter of the arrival process on the required bandwidth. When varying the activity parameter, we also vary the number of sources so that the total mean rate of arrivals keeps constant. The goal here is to find out if the network operator may be able, once the allocated bandwidth, to use it efficiently whatever the profile of the connectionless traffic. Here the mean burst duration is constant and equals 200 ms and we vary the silence intervals duration.

The main conclusion which can be drawn from this figure is that the FC and FCDA mechanisms seem to be almost insensitive to the activity parameter of the sources. This is a very interesting feature since, if the activity parameter of typical sources for connectionless services is expected to be very low, it is also expected to vary a lot from one type of source to another.

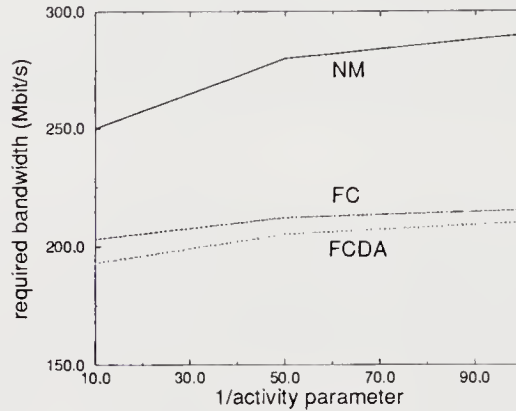


Figure 8 Influence of the activity parameter.

Even if, from the two preceding experiments, some interesting features of the FCDA mechanism have been highlighted, its gain with respect to FC mechanism has not revealed considerable with respect to the FC mechanism.

However, until now, these mechanisms were only compared in a context where, because of the choice of the rate of the output link of CLS A, not many control requests were imposed to CLS B and C. These conditions are now changed by adding a background source which feeds CLS A. The characteristics of this source are a peak rate of 120 Mbit/s and an activity parameter of 0.1. This of course increases the load of CLS A.

Figure 9 shows the obtained results. Only FC and FCDA mechanisms are compared, but two cases are considered for FCDA mechanism, depending on the assumed probability of refusals of allocation requests. FCDA 1 corresponds to the same case as before, i.e. $P_f = 0.1$. FCDA 2 corresponds to $P_f = 0.05$.

The required bandwidth is of course greater than in the previous situation, where no external traffic was imposed to CLS A. It can also be observed that the gain obtained for the FCDA mechanism with respect to the FC mechanism becomes significant in these conditions. It is of 15 % for the same conditions of refusals as before and even of 25 % in the case of lower probability of refusals. This probability is difficult to characterize, since it corresponds to the capability of ATM network, in view of its load, to allocate more bandwidth to the CLSs output links. This is mainly related to the CAC function used by the network. However the cases investigated here do not seem to us exceedingly optimistic ; this allows to hope that in a real case the proposed mechanism will induce even better performances.

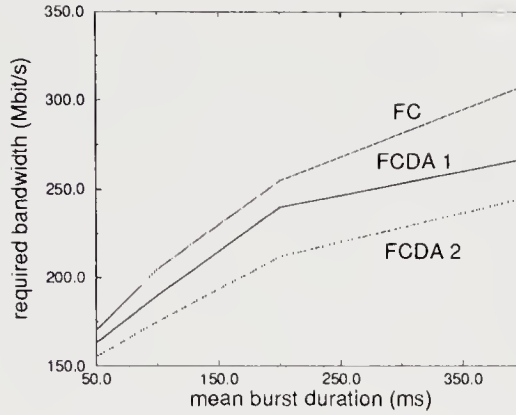


Figure 9 Influence of load on CLS A.

Influence of the distance

Until now, we have only studied the case where the buffers of the sources were supposed to have an infinite capacity. In this paragraph we remove this restriction and, in particular, we analyze the influence of the network size on the required buffer sources. Indeed, it is desirable that these buffer requirements are less sensitive with respect to the network size.

Figure 10 shows the influence between sources and CLS and between CLSs in the case where the bandwidth in the FC case and the initial bandwidth in the FCDA case correspond to the minimum values required to guarantee the required QoS. The sources have 100 ms mean burst sizes and the global load remains as in the previous sections.

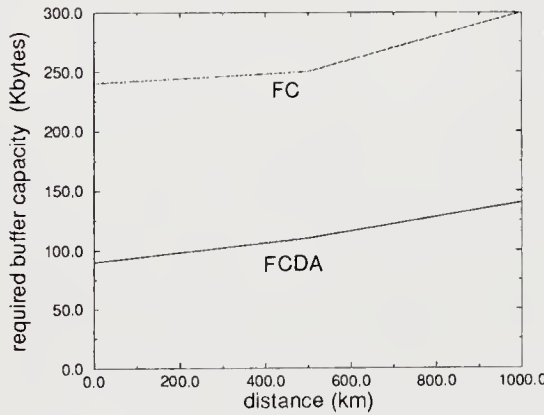


Figure 10 Influence of distance on sources buffers.

We conclude that the proposed mechanism allows the use of smaller buffers for the sources and reduced the sensitivity of these buffer sizes with respect to the size of the network.

Transient behaviour

In order to better understand the impact of the mechanism, we show in the next figures a typical transient behaviour of the FC and FCDA mechanisms in a situation of congestion. Figure 11 shows the evolution of the occupancy of the buffer of CLS B during a typical period of congestion. At the beginning of the represented period, the CLS has made a request for bandwidth allocation which has been accepted. The output link rate becomes $255 \text{ Mbit/s}^\dagger$ instead of the former 225 Mbit/s . As we see on this figure, this allocation does not prevent the buffer occupancy from reaching the threshold T_{ctrl} (900 packets). However, this allocation allows a shorter period of control of the sources. Moreover, we observe that, when using the FC mechanism, the period of control is followed by another one. This is because of the release of controlled sources, which have accumulated a lot of packets while being controlled. This second period of control leads to the saturation of two sources buffers and to losses in one of these buffers (See Figure 13).

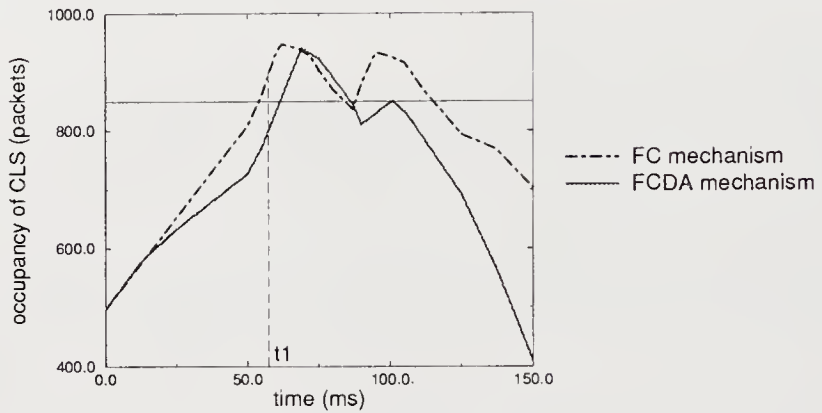


Figure 11 Buffer occupancy of CLS B.

This phenomenon is avoided in the FCDA for one main reason. This is that, when sources are released, the CLS disposes of a larger output rate, thus avoiding a second period of “high congestion”. Other reasons are, first, that, as we underlined it, the period of control is slightly shorter with this mechanism, so that sources are less stressed than with the FC mechanism; second, observe on Figure 12 that, even when using flow control, the FCDA mechanism reduces less the rate of sources than the FC mechanism. This is because the coefficient of reduction of the rate of sources is function of the output rate of the congested CLS (since the objective of the flow control mechanisms we study is to reach a target load ρ). Note the slight delay (corresponding to the propagation delay) between the time of the control decision (t_1) and the time of the effective reduction of the rate of the sources (t_2) (See Figures 11 and 12).

The direct effects of the flow control on packet losses in sources is made clear in Figure 13. On this figure, the occupancy of the buffers of three of the sources is represented,

[†]We suppose ATM interfaces at 622 Mbit/s . Let us recall here that only 155 Mbit/s and 622 Mbit/s interfaces have been defined by the ITU.

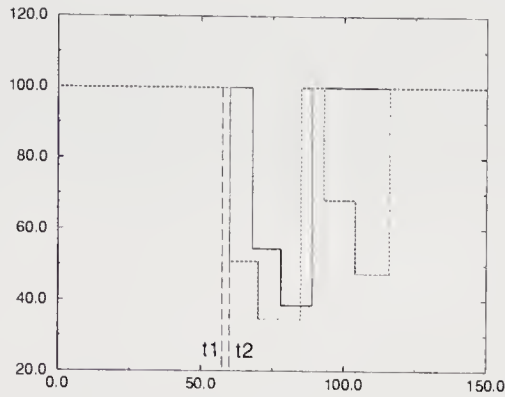


Figure 12 Source rates.

during the same period as on the above figures. There is no loss in the FCDA mechanism whereas the two periods of control cause a progressive saturation of two of the sources (one of which remains saturated even after the observed period) and cause losses in one.

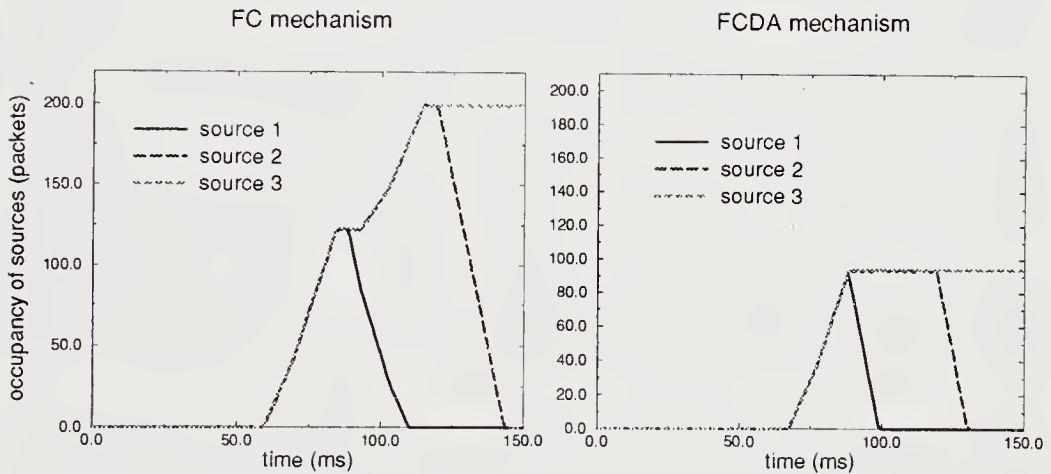


Figure 13 Source buffers (FC and FCDA).

4 CONCLUSIONS

We propose a traffic management mechanism combining flow control and dynamic bandwidth allocation in connectionless networks using an ATM network infrastructure. The mechanism allows a gain in the utilization of the capacity of the ATM network and reduces the sensitivity of the performances of the connectionless network with respect to the traffic characteristics. The next steps of our study are as follows. First, we are taking

into account the usage parameter control that has been standardized by the ETSI for the CBDS (Connectionless Broadband Data Service) service. In such a case, the backpressure flow control to the sources used in the present study should not constrain the input traffic under the negotiated traffic contract. Second, we are analyzing the performances of a connectionless network on top of an ATM network offering the ABR transfer capability in order to compare the results with the performances of the system analyzed in the present paper.

5 REFERENCES

- ITU Recommendation I.211, B-ISDN Service aspects.
- Derman, C. (1970) *Finite State Markovian Decision Processes*. Academic Press, New York.
- Gallassi, G., Gerla, M. and Tai, T.Y. (1992) LAN/MAN interconnection to ATM: a simulation study. Proceedings of INFOCOM92, 2270-9.
- Heijenk, G.J. and Niemegeers, I.G. (1992) Variable bandwidth connections for a connectionless service on ATM - Performance modelling and evaluation. IFIP Transactions C (Communications Systems), Volume C-4, 361-72.
- Mongiovi, L., Farrell, M. and Trecordi, V. (1991) A proposal for interconnecting FDDI networks through B-ISDN. Proceedings of INFOCOM91, volume3, 1160-7.
- Puterman, M.L. (1994) *Markov Decision Processes*. Wiley Series in Probability and Mathematical Statistics.
- Roberts, J.W. (1991) Variable-Bit-Rate Traffic Control in B-ISDN. IEEE Communications Magazine, September 1991.
- Serfozo, R. (1979) An equivalence between continuous and discrete time Markov decision processes. Operation Research 27, 616-20.
- Van den Berg, H. and Smeitink, E. (1995) Design and dimensioning of an ATM based connectionless overlay network. 11th European Network Planning Workshop.
- Vickers, B.J. and Suda, T. (1994) Connectionless service for public ATM networks. IEEE Communications Magazine, August 94, 34-42.
- Yamamoto, M. *et al.* (1993) Traffic control scheme for interconnection of FDDI networks through ATM network. Proceedings of INFOCOM93, 411-20.

Serge Halberstadt graduated from Ecole Polytechnique in 1992 and obtained the Diplôme d'Etudes Approfondies in Probability from University Paris XI in 1993. He is a doctorate student at the Networks Department of Ecole Nationale Supérieure Nationale des Télécommunications. His research interests are in traffic management in ATM networks, queuing theory and Markov decision processes.

Daniel Kofman received the M.Sc. and Ph.D. degrees from Ecole Nationale Supérieure Nationale des Télécommunications (ENST-Paris). He is an Associate Professor at the Networks Department of the ENST, where he teaches and conducts research in the field of high speed networks. His main research interest is currently traffic management in ATM networks. He is the co-author of a book entitled "Réseaux Haut Débit, réseaux ATM et

réseaux locaux" (Inter-Editions) and he was the CO-Chairman of the "First Workshop on ATM Traffic Management" IFIP WG6.2.

Annie Gravey passed the Agrégation de Mathématique in 1978 and in June 1981, she received her doctor's degree in Signal Theory and Automatics from the University of Paris-Sud, Orsay. In 1981, she joined the CNET (National Telecommunications Research Center) in Lannion. Her current interests include queueing theory and performance evaluation of ATM systems.

ATM traffic prediction using FIR neural networks

Z. Fan and P. Mars

*School of Engineering, University of Durham,
South Road, Durham, DH1 3LE, UK.*

Tel: +44 191 3742559 Fax: +44 191 3743838

Email: (zhong.fan, philip.mars)durham.ac.uk

Abstract

ATM networks support a wide range of multimedia traffic. Various BISDN VBR sources generate traffic at significantly different rates. The traffic can often have time-varying characteristics which are not well understood currently. However, traffic management techniques require traffic parameters that can capture the various traffic characteristics and adapt to the changing network environment. In this paper, we present a novel neural network approach to characterize and predict the complex arrival process. The FIR multilayer perceptron model and its training algorithm are discussed in this paper. It is shown that the FIR neural network can adaptively predict the traffic by learning the relationship between the past and the future traffic variations. Based on the experimental results, we conclude that the FIR neural network is an attractive tool for traffic prediction and hence has an excellent potential for use in some congestion control schemes.

Keywords

ATM, traffic prediction, FIR neural networks

1 Introduction

Asynchronous Transfer Mode(ATM) has been recommended by CCITT as the transfer mode for the future broadband ISDN(BISDN). ATM networks are expected to support a diverse set of applications, such as data, voice and video, each having different traffic characteristics. Accurate characterization of the multimedia traffic is essential in order to develop a robust set of traffic descriptors. Such a set is required by the Usage Parameter Control(UPC) algorithm for traffic enforcement and the Connection Admission

Control(CAC) algorithm for bandwidth allocation utilizing the statistical multiplexing gain. However, for the time being, there are no comprehensive measurements that permit designers to satisfactorily address the characteristics of various communication services in a realistically accurate manner. This is especially true for Variable Bit Rate(VBR) traffic.

During the duration of a connection, the period at which a source generates traffic is referred to as an *active* period, whereas a *silent* period corresponds to the time between the active periods during which no traffic is generated. Traffic generated by a VBR source either alternates between the active and silent periods, or is a continuous bit stream with varying rates. This traffic is highly bursty and correlated(in comparison to a Poisson process). Burstiness can be defined by the ratio of the peak bit rate to average bit rate or the squared coefficient of variation of the interarrival times of cells, c_1^2 (variance divided by the square of the mean). For example, c_1^2 for the packet arrival process from a single voice source is 18.1, while c_1^2 for a Poisson process is 1 [Sriram 86, Heffes 86]. Although the aggregate packet arrival process with many components does behave like a Poisson process over relatively short time intervals, under heavy loads the congestion in the multiplexer is determined by the behaviour of the arrival over much longer time intervals, where it does not behave like a Poisson process. Accordingly, characterization of traffic from VBR sources is very difficult.

As mentioned above, the congestion control schemes(e.g., CAC and UPC) in ATM networks require specific knowledge of the statistical behaviour of the input traffic declared via its traffic descriptors. Parameters such as peak bit rate, average bit rate, and burst length are often used as a simple set of parameters characterizing the traffic. More complicated second-order time domain parameters(e.g., IDI, IDC) are also used to capture the burstiness and the correlation properties of the arrival stochastic process especially those of VBR video and voice sources [Habib 92]. In [Heffes 86], the aggregate arrival process from N voice sources is approximated by a nonrenewal process, i.e., a two-state Markov Modulated Poisson Process(MMPP). In [Daigle 86], very complex mathematical models such as semi-Markov process and continuous-time Markov chain are used to characterize the voice traffic. Traffic descriptors using simple parameters will not accurately characterize very rapid changes in the bit rate time variations of the traffic over short intervals and often ignore the bursty nature of the traffic. On the other hand, those mechanisms using more sophisticated parameters are computationally expensive and impractical.

To solve this problem, a neural network based traffic prediction approach is proposed in this paper. The neural network can predict the bit rate variations of a complex stochastic process and capture the probability density function(pdf) of the traffic. It is shown that the neural prediction is accurate enough to characterize the actual traffic and can be used in policing and CAC functions. It can also be used in some feedback control schemes. It has been argued that traditional reactive congestion control is not suitable for ATM networks due to the effects of high-speed channels. Recently, Amenyo et al. [Amenyo 91] have proposed a new congestion control scheme called proactive control. Underlying its feasibility and effectiveness are traffic predictions of correlated input traffic streams into network nodes. These predictions are used to obviate the problem of propagation delays.

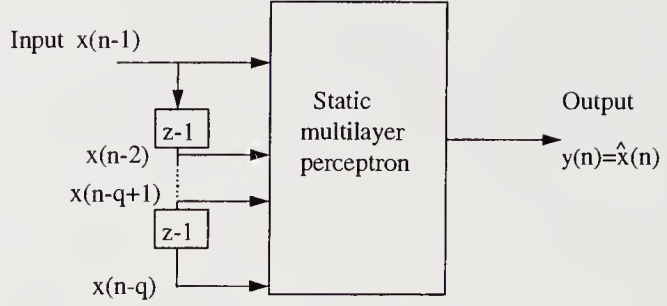


Figure 1: Static multilayer perceptron used as a nonlinear predictor.

So we can apply our neural prediction method to this framework as well.

2 FIR Neural Network

It is well known that neural networks are capable of performing nonlinear mappings between real-valued inputs and outputs. A three-layered feedforward neural network (multilayer perceptron), with sigmoidal units in the hidden layer, is able to approximate an arbitrary nonlinear function to any desired degree of accuracy [Funahashi 89, Hornik 89]. This kind of neural network is trained with the backpropagation(BP) algorithm. One limitation of the standard BP algorithm is that it can only learn an input-output mapping that is *static*. This form of static input-output mapping is well suited for pattern recognition applications, where both the input and output vectors represent *spatial* patterns that are independent of time [Haykin 94].

The standard BP algorithm may also be used to perform nonlinear prediction on a stationary time series [Lapedes 87]. We may use a static multilayer perceptron, as depicted in Figure 1, where the input elements labeled z^{-1} represent unit delays. The input vector \mathbf{x} is defined in terms of the past samples $x(n-1), x(n-2), \dots, x(n-q)$ as follows:

$$\mathbf{x} = [x(n-1), x(n-2), \dots, x(n-q)]^T \quad (1)$$

where q is the prediction order. Thus the scalar output $y(n)$ of the multilayer perceptron equals the one-step prediction $\hat{x}(n)$, as shown by

$$y(n) = \hat{x}(n) \quad (2)$$

The actual value $x(n)$ of the input signal represents the desired response.

However, if we want to capture the *dynamic* properties of the time-varying signals, we have to extend the design of a multilayer perceptron so as to represent *time* in it. One of the methods is the so-called Time Delay Neural Network(TDNN), which was first used in

[Lang 88] to perform speech recognition. The TDNN is a multilayer feedforward network in which the outputs of a layer are buffered several time steps and then fed fully connected to the next layer. It was devised to capture explicitly the concept of time symmetry as encountered in the recognition of an isolated phoneme using a spectrogram.

The TDNN topology is in fact embodied in a multilayer perceptron in which each synapse is represented by a Finite Impulse Response (FIR) filter. This latter neural network is referred to as a FIR multilayer perceptron, which can be trained with an efficient algorithm called *temporal backpropagation* [Wan 94]. It can be shown that the TDNN and the FIR network are functionally equivalent. However, the FIR network is more easily related to a standard multilayer network as a simple temporal or vector extension. The FIR representation also leads to a more desirable adaptation scheme. So in this paper, we adopt this kind of FIR network as our traffic predictor.

2.1 FIR Network Model

As mentioned above, the traditional model of a multilayer perceptron forms a static mapping; there are no internal dynamics. A modification of the basic neuron is accomplished by replacing each synaptic weight by a FIR linear filter. By FIR we mean that for an input excitation of finite duration, the output of the filter will also be of finite duration. For this filter, the output $y(k)$ equals a weighted sum of past delayed values of the input:

$$y(k) = \sum_{n=0}^T w(n)x(k-n) \quad (3)$$

On the basis of Eq. 3, we may formulate the model of a FIR neuron as follows. Let $w_{ji}(l)$ denote the weight connected to the l th tap of the FIR filter modeling the synapse that connects the output of neuron i to neuron j ($i = 1, 2, \dots, p$). The index l ranges from 0 to M , where M is the total number of delay units built into the design of the FIR filter. Let $y_j(n)$ denote the output signal of neuron j and $x_i(n)$ the input signal. Hence, we have

$$v_j(n) = \sum_{i=1}^p \sum_{l=0}^M w_{ji}(l)x_i(n-l) - \theta_j \quad (4)$$

$$y_j(n) = \varphi(v_j(n)) \quad (5)$$

where $v_j(n)$ is the net activation potential of neuron j , θ_j is the externally applied threshold and $\varphi(\cdot)$ is the nonlinear activation function of the neuron.

We may rewrite Eq. 4 and Eq. 5 in matrix form by introducing the following definitions for the state vector and weight vector for synapse i , respectively:

$$\mathbf{x}_i(n) = [x_i(n), x_i(n-1), \dots, x_i(n-M)]^T \quad (6)$$

$$\mathbf{w}_{ji} = [w_{ji}(0), w_{ji}(1), \dots, w_{ji}(M)]^T \quad (7)$$

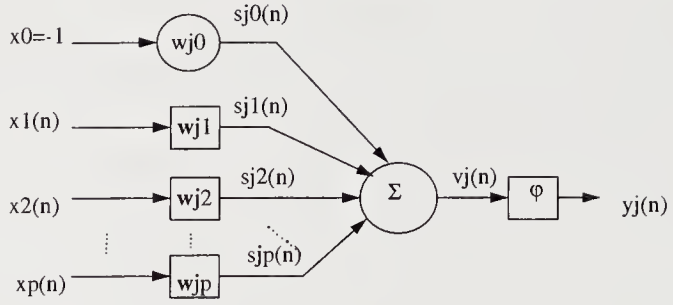


Figure 2: Dynamic model of a neuron, incorporating synaptic FIR filters.

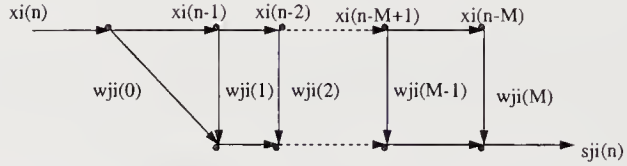


Figure 3: Signal-flow graph of a synaptic FIR filter.

We may thus express the output $y_j(n)$ of neuron j by the following equation:

$$y_j(n) = \varphi\left(\sum_{i=1}^p \mathbf{w}_{ji}^T \mathbf{x}_i(n) - \theta_j\right) \quad (8)$$

This FIR model of a single artificial neuron is shown in Figure 2, where the weight w_{j0} connected to the fixed input $x_0 = -1$ represents the threshold θ_j . The signal-flow graph representation of a FIR filter is shown in Figure 3.

We may construct a multilayer perceptron whose hidden and output neurons are all based on the above FIR model. Such a neural network structure can be referred to as a FIR multilayer perceptron. The difference between the FIR multilayer perceptron and the standard one is that the static forms of the synaptic connections between the neurons in the various layers of the network are replaced by their dynamic versions (i.e., scalars are replaced by vectors and multiplications by vector products).

2.2 Temporal Backpropagation Learning

Given an input sequence $x(k)$, the network produces the output sequence $y(k) = \mathcal{N}[W, x(k)]$, where W represents the set of all filter coefficients in the network. Define the instantaneous error $e^2(k) = \|d(k) - y(k)\|^2$ as the squared Euclidean distance between the network output $y(k)$ and the desired output $d(k)$. Therefore the objective of training corresponds to minimizing over W the cost function:

$$C = \frac{1}{2} \sum_{k=1}^K e^2(k)$$

where the sum is taken over all K points in the training sequence. In [Wan 94], an algorithm called temporal backpropagation is proposed to minimize C . The weight-update equation is shown by the following pair of relations:

$$\mathbf{w}_{ji}(k+1) = \mathbf{w}_{ji}(k) - \eta \frac{\partial C}{\partial v_j(k)} \frac{\partial v_j(k)}{\partial \mathbf{w}_{ji}(k)} = \mathbf{w}_{ji}(k) + \eta \delta_j(k) \mathbf{x}_i(k) \quad (9)$$

$$\delta_j(k) = \begin{cases} e_j(k) \varphi'(v_j(k)), & \text{neuron } j \text{ in the output layer} \\ \varphi'(v_j(k)) \sum_{m \in \mathcal{A}} \Delta_m^T(k) \mathbf{w}_{mj}, & \text{neuron } j \text{ in a hidden layer} \end{cases} \quad (10)$$

where η is the learning-rate parameter, \mathcal{A} is defined as the set of all neurons whose inputs are fed by neuron j in a forward manner and $\Delta_m(k)$ is defined as follows:

$$\Delta_m(k) = [\delta_m(k), \delta_m(k+1), \dots, \delta_m(k+M)]^T \quad (11)$$

It is obvious that the above equations represent a *vector generalization* of the standard backpropagation algorithm. In fact, if we replace the input vector $\mathbf{x}_i(n)$, the weight vector \mathbf{w}_{mj} , and the local gradient vector Δ_m by their scalar counterparts, the temporal backpropagation algorithm reduces to the standard backpropagation for static networks. To calculate $\delta_j(k)$ for a neuron j located in a hidden layer, we filter the δ 's from the next layer backwards through the FIR synapses for which the given neuron feeds (see Figure 4). Thus δ 's are formed not by simply taking weighted sums, but by backward filtering. For each new input and desired response vector, the forward filters are incremented one time step and the backward filters one time step. The weights are then adapted on-line at each time increment.

Temporal backpropagation preserves the symmetry between the forward propagation of states and the backward propagation of error terms. The sense of parallel distributed processing is thereby maintained. Furthermore, each unique weight of synaptic filter is used only once in the computation of the δ 's; there is no redundant use of terms experienced in the instantaneous gradient model.

However, careful inspection of the above equations reveals that the calculations for the $\delta_j(k)$'s are noncausal. We may formulate the causal form of the temporal backpropagation algorithm by a simple reindexing:

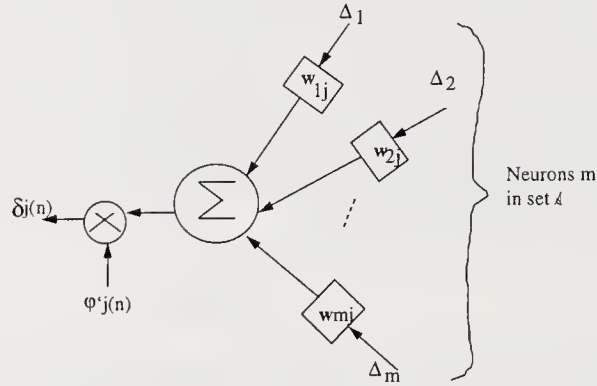


Figure 4: Backpropagation of local gradients through an FIR multilayer perceptron.

For neuron j in the output layer, compute

$$w_{ji}(k+1) = w_{ji}(k) + \eta \delta_j(k) x_i(k) \quad (12)$$

$$\delta_j(k) = e_j(k) \varphi'_j(k) \quad (13)$$

For neuron j in a hidden layer, compute

$$w_{ji}(k+1) = w_{ji}(k) + \eta \delta_j(k-lM) x_i(k-lM) \quad (14)$$

$$\delta_j(k-lM) = \varphi'(v_j(k-lM)) \sum_{m \in A} \Delta_m^T(k-lM) w_{mj} \quad (15)$$

where M is the total synaptic filter length, and the index l identifies the hidden layer in question. Specifically, $l = 1$ corresponds to one layer back from the output layer; $l = 2$, two layers back from the output layer; and so on.

3 ATM Traffic Prediction Using FIR Networks

Neural networks have adaptation capability that can accommodate nonstationarity. Their generalization capability makes them flexible and robust when facing new and noisy data patterns. Once the training is completed, a neural network can be computationally inexpensive even if it continues to adapt on-line. Actually, neural networks have been used in call control, switch control and routing [Morris 94, Hiramatsu 90]. Here we use a FIR neural network as a multimedia traffic predictor in ATM networks. The role of the neural network is to capture the unknown complex relation between the past and future values of the traffic.

3.1 Predictor Training Configuration

Consider a scalar time series denoted by $x(n)$, which is described by a nonlinear regressive model of order q as follows [Haykin 94]:

$$x(n) = f(x(n-1), x(n-2), \dots, x(n-q)) + \varepsilon(n) \quad (16)$$

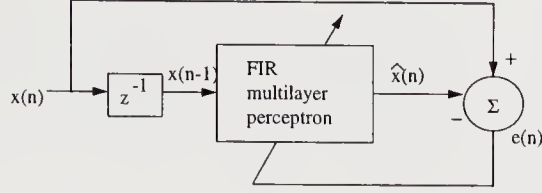


Figure 5: Training scheme of the FIR network.

where f is a nonlinear function of its arguments and $\varepsilon(n)$ is a residual. It is assumed that $\varepsilon(n)$ is drawn from a white Gaussian process. The nonlinear function f is unknown, and the only thing that we have available to us is a set of observables: $x(1), x(2), \dots, x(N)$, where N is the total length of the time series. We may use a FIR multilayer perceptron as a one-step predictor of some order q to model the time series, as shown in Figure 5. Specifically, the network is designed to make a prediction of the sample $x(n)$, given the past q samples $x(n-1), x(n-2), \dots, x(n-q)$, as shown by

$$\hat{x}(n) = F(x(n-1), x(n-2), \dots, x(n-q)) + e(n) \quad (17)$$

The nonlinear function F is the approximation of the unknown function f , which is computed by the FIR multilayer perceptron. The actual sample value $x(n)$ acts as the desired response. Hence the FIR multilayer perceptron is trained so as to minimize the squared value of the prediction error:

$$e(n) = x(n) - \hat{x}(n), \quad q+1 \leq n \leq N. \quad (18)$$

In the neural network literature the above training scheme is referred to as *teacher forcing*, while in the control and signal processing literature, it is referred to as *equation-error adaptation*.

In our application, the FIR multilayer perceptron is designed as a 1-5-1 fully connected feedforward network with 3:3 taps per layer. Selection of these dimensions is based mostly on trial and error. In general, selection of dimensions for neural networks remains an open question in need of further research. The FIR network is trained with the causal form of temporal backpropagation and the *mean-squared error* (MSE) is used as a performance measure. To increase the rate of learning and yet avoid the danger of instability, a momentum term is added to the weight-update equation, i.e.,

$$\Delta \mathbf{w}_{ji}(k) = \alpha \Delta \mathbf{w}_{ji}(k-1) + \eta \delta_j(k) \mathbf{x}_i(k) \quad (19)$$

where α is a positive number called the momentum constant. The learning rate η and momentum constant α are set at 0.1 initially. It has been found that the BP learning algorithm may learn faster when the sigmoidal activation function built into the neuron model of the network is asymmetric than when it is nonsymmetric. So we adopt the hyperbolic tangent activation function in the hidden layer, which is defined by

$$\varphi(v) = a \tanh(bv)$$

where $a = 1.716$ and $b = 2/3$. In some of our experiments, we have also used some heuristics to accelerate the convergence of backpropagation learning through learning rate adaptation [Haykin 94]. Simulations have been performed to obtain the neural network data set for both training and testing(cross-validation) purposes. Since we use the logistic function $\varphi(v) = 1/(1 + \exp(-v))$ for the output neuron, we have to normalize the traffic data so that all the values fall between 0 and 1.

3.2 Traffic Models

In this section, we briefly describe the models for video arrival process and voice arrival process used in our experiments.

3.2.1 Video Arrival Process Model

Video is presented to users as a series of frames in which the motion of the scene is reflected in small changes in sequentially displayed frames. Video frames are generated at a constant rate defined by the playout rate. As the amount of data transmitted per frame varies due to intraframe and interframe coding, video applications generate traffic in a continuous manner at varying rates. Video is a relatively new service in communication networks and its traffic characteristics are not well understood. It is also quite different from voice or data in that its bit streams exhibit various types of correlations between consecutive frames.

The characteristics of the video signal depends primarily on two factors: 1) the nature of the video scene, and 2) the type of VBR coding technique employed(e.g., motion-compensated discrete cosine transform, interframe DPCM, etc.). For the purpose of simplicity, in this paper, we focus on video services with uniform activity level scenes, i.e., the change in the information content of consecutive frames is not significant [Onvural 94]. A typical application of this type is video telephone where the screen shows a person talking. In general, correlations in video services with uniform activity levels last for a short duration and decay exponentially with respect to the time. The simulation model used to generate this kind of video coded traffic is a continuous-state discrete-time stochastic process. A first-order autoregressive(AR) Markov model is proposed in [Maglaris 88], which estimates the bit rate at the n th frame from the bit rate at the $(n - 1)$ st frame to be

$$\lambda(n) = a\lambda(n - 1) + bw(n) \quad (20)$$

where $\lambda(n)$ denotes the bit rate of the n th frame in bits/pixel, a and b are constants and $w(n)$ is a Gaussian random variable with mean m and variance 1. There are about 250000 pixels per frame and 30 frames/s, thus 1 bit/pixel corresponds to 7.5 Mbits/s. The mean $E(\lambda)$, and the autocovariance of the bit rate $C(n)$ are equal to

$$E(\lambda) = bm/(1 - a) \quad (21)$$

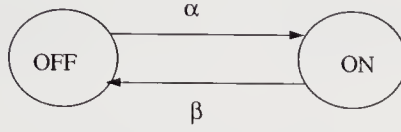


Figure 6: IPP model.

$$C(n) = b^2 a^n / (1 - a^2) \quad (22)$$

which can be used to determine the two unknown variables a , b and m :

$$a = 0.8781, \quad b = 0.1108, \quad m = 0.572 \quad (23)$$

The model is found to be quite accurate compared with the actual measurements and is suitable for simulation studies.

3.2.2 Voice Arrival Process Model

A voice source alternates between talk spurts(active) and silent periods. To achieve higher resource utilization, a speech activity detection may be used at the VBR voice source so that voice packets are generated only when the source is active, thereby, increasing the transmission efficiency. The correlated generation of voice packets within a call can be modeled by an Interrupted Poisson Process(IPP). In an IPP model, each voice source is characterized by ON(corresponding to talk spurt) and OFF(corresponding to silence duration) periods, which appear in turn. During the ON period, the interarrival times of packets are exponentially distributed(i.e., in a Poisson manner), while no packets are generated during the OFF period. The transition from ON to OFF occurs with the rate β , and the transition from OFF to ON occurs with the rate α (see Figure 6). Hence the ON and OFF periods are exponentially distributed with the mean $1/\beta$ and $1/\alpha$. To specify this model completely, we assume that the packet-generation rate during the active period is 32kbps, the mean talk spurt is $1/\beta = 352\text{ms}$ and the mean silence period is $1/\alpha = 650\text{ms}$.

4 Simulations

In this section, we demonstrate the effectiveness of the neural network used as a traffic predictor. Extensive simulations have been performed. The packet arrival process is generated from packetized video sources or/and packetized voice sources according to the models discussed in the previous section. We have used different data for the training

FIR network	MSE for the traing set	MSE for the test set
1-5-1	0.00414	0.00423
1-10-1	0.00410	0.00415

Table 1: MSE of the experiment for video traffic.

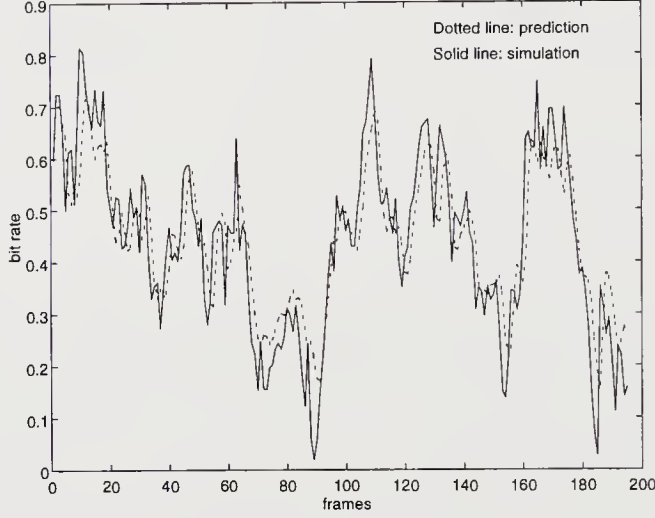


Figure 7: Prediction results for the bit rate of the video traffic.

sets and test sets by choosing different initial values of the arrival process or different seeds of the random number generator. For example, for the video arrival process, we have generated 2000 traffic data elements, starting with $\lambda(0) = 0.6$. The first 400 elements have been chosen to be the training set, and the next 1600 to be the test set. We have also tried a more complicated 1-10-1 network model for the same data sets, but no significant performance improvement is observed. The values of MSE of the above experiment are summarized in Table 1. Other prediction results for the test sets will be reported in the following.

Experiment 1: In this experiment, we use three video sources. The FIR network is used to predict the bit rate of the superposition video arrival process over the next frame. Therefore the lag time is $1/30$ sec, which is the frame generation rate. The prediction results are shown in Figure 7 and Figure 8, illustrating that the predicted traffic has almost the same statistical characteristics as those of the actual traffic.

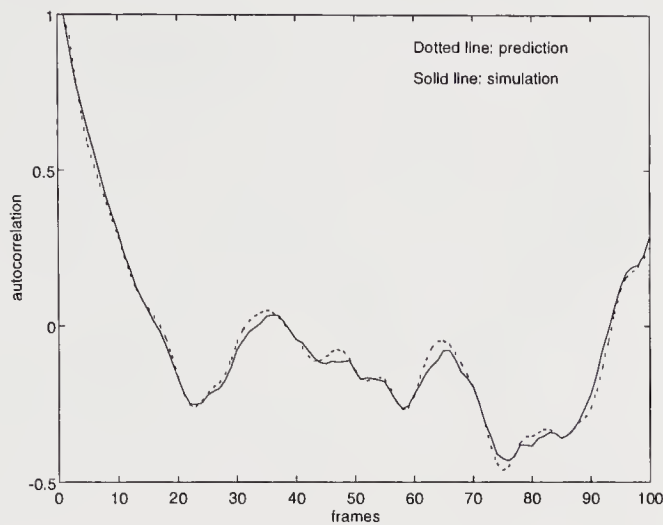


Figure 8: Comparison of the autocorrelation function of the predicted video traffic and the actual one.

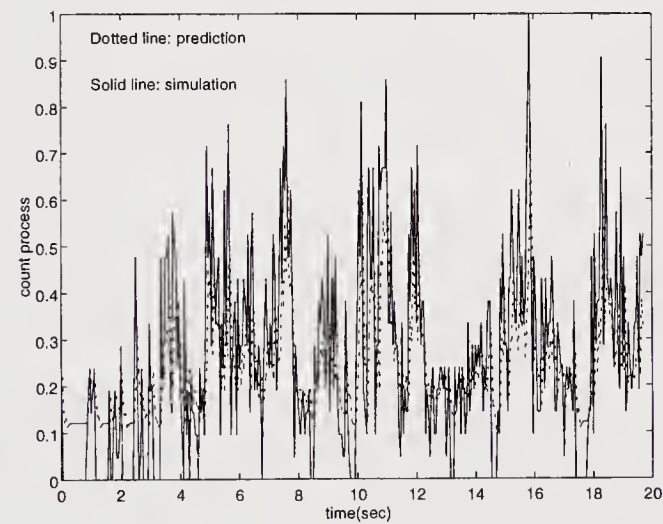


Figure 9: Prediction results for the arrival process of voice sources.

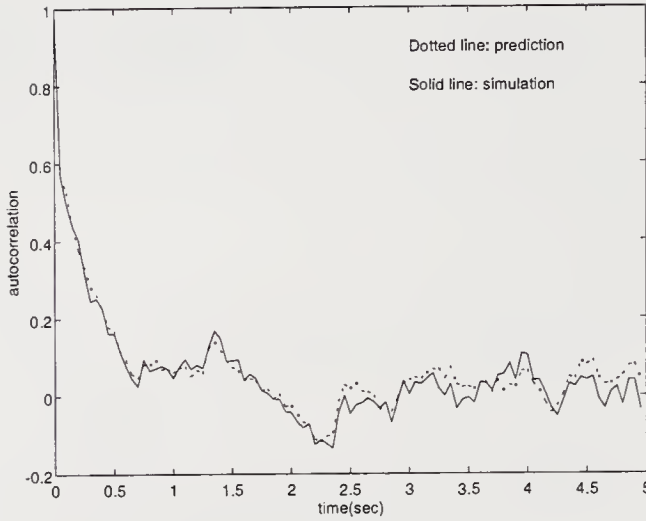


Figure 10: Comparison of the autocorrelation of the number of packet arrivals of the predicted voice traffic and the actual one.

Experiment 2: In this experiment, we use three voice sources. Here the time series $x(n)$ is used to represent the count process $N(0, t)$ which measures the number of packet arrivals in time $(0, t)$. The arrival process is sampled at every sampling period T_s . The choice of the parameter T_s is influenced by the type of the traffic and should guarantee that the used sampled version of the arrival process captures all correlations contained in the actual process. In this application T_s has been found to be 50 ms [Tarraf 94]. Figure 9 and Figure 10 show that the neural network prediction is very close to the actual traffic values.

Experiment 3: In this experiment, one video source and three voice sources are used to generate a heterogeneous superposition arrival process. Prediction results of the count process are shown in Fig. 11 and Fig. 12. It should be pointed out that more training iterations of the neural network are needed in this experiment than in the previous ones, since the heterogeneous traffic is more difficult to characterize.

Experiment 4: Recently, Leland et al. demonstrated that Ethernet local area network traffic is statistically *self-similar* [Leland 93]. To capture this *fractal* behaviour, they proposed to model the traffic using deterministic *chaotic* maps. Chaos is a dynamical system phenomenon in which simple, low order, nonlinear deterministic equations can produce behaviour that mimics random processes. To illustrate the underlying idea, consider a nonlinear map $f(\cdot)$ that describes the evolution of a state variable $x(n) \in (0, 1)$ over discrete time as $x(n+1) = f(x(n))$. The packet generation process for an individual

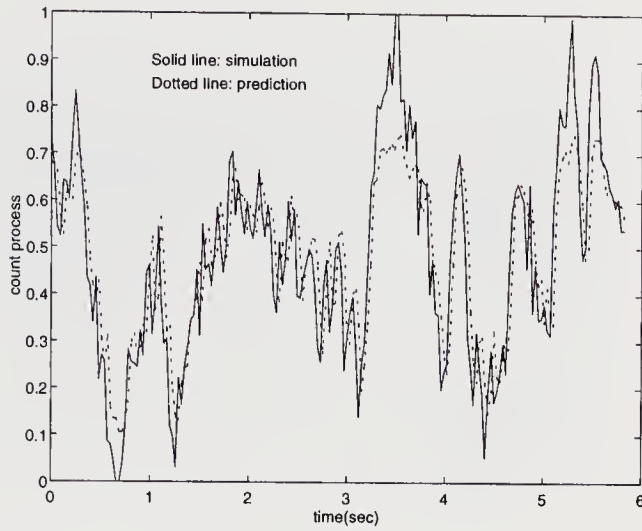


Figure 11: Prediction results for the heterogeneous traffic.

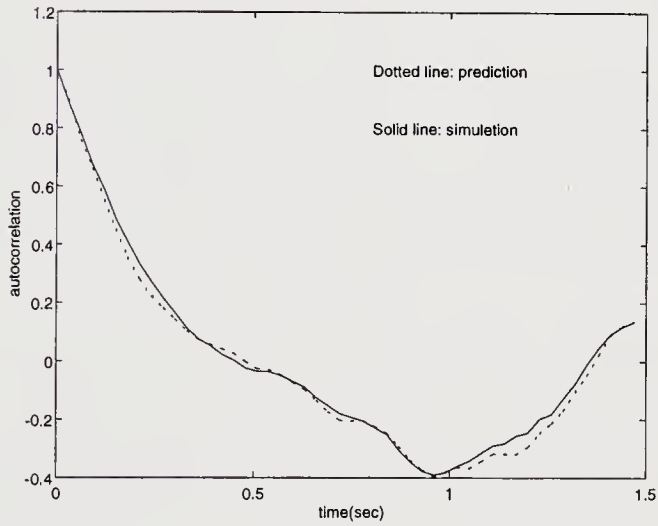


Figure 12: Comparison of the autocorrelation function of the predicted traffic and the actual one in experiment 3.

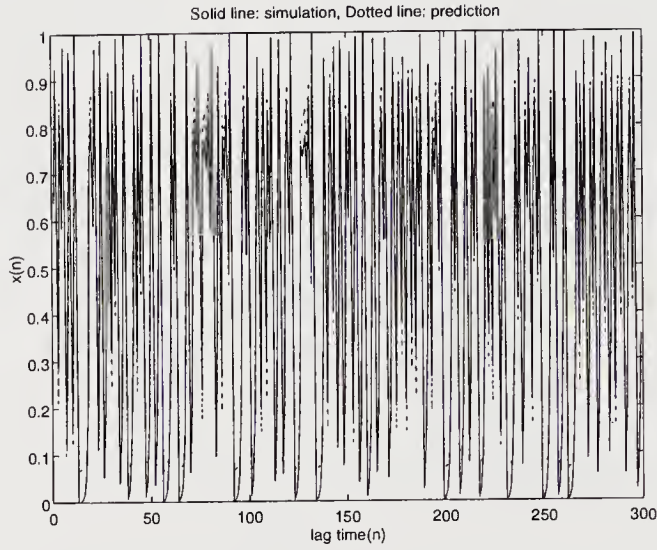


Figure 13: Prediction results for the training set of the chaotic time series.

source can now be modeled by stipulating that the source generates one or no packet at time n depending on whether $x(n)$ is above or below an appropriately chosen threshold. If f is a chaotic map, the resulting packet process can mimic complex packet traffic phenomena. Once an appropriate chaotic map has been derived from a set of traffic measurements, generating a packet stream for an individual source is generally quick and easy. On the other hand, deriving an appropriate nonlinear chaotic map based on a set of actual traffic measurements currently requires considerable guessing and experimenting. Nevertheless, studying arrival streams to queues that are generated by nonlinear chaotic maps may well provide new insight into the performance of queueing systems where the arrival processes exhibit fractal properties.

Here as another experiment, we train the FIR network to perform one-step prediction of a chaotic time series. A chaotic time series generated by the so called logistic map is defined as [Rasband 90]

$$x(n+1) = 4x(n)(1-x(n)) \quad (24)$$

where the values of $x(n)$ are all in the range $(0, 1)$. The prediction results for the training and test data sets are encouraging, as shown in Figure 13 and Figure 14 respectively.

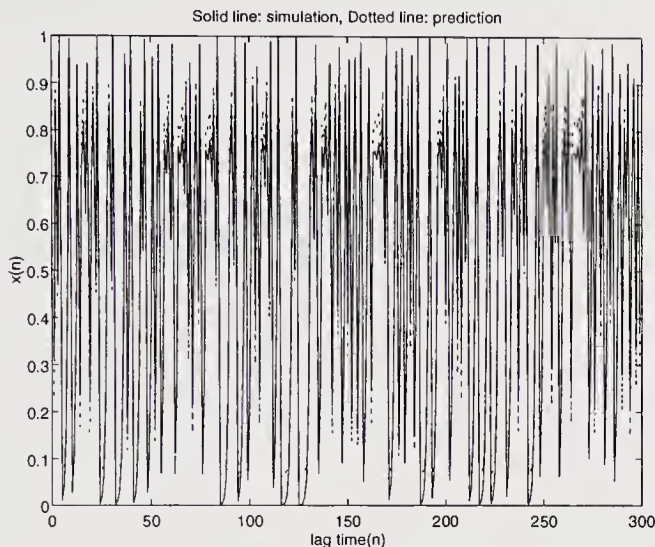


Figure 14: Prediction results for the test set of the chaotic time series.

5 Conclusion

In this paper, we have shown that a FIR network constitutes a powerful tool for use in ATM traffic prediction. The theoretical justification of this approach is that neural networks are capable of approximating any continuous function and perform non-parametric regression. Furthermore, a FIR neural network extends the standard multilayer perceptron to a temporal processing version which is more suitable for modeling of time series. After completing the training phase of the neural network, it can successfully learn the actual pdf of the offered traffic (instead of the approximated simple parameters, such as the peak and mean bit rates). Hence the neural network can be used as an effective traffic descriptor.

In ATM networks, traffic management techniques require traffic parameters that can capture the various traffic characteristics and adapt to the changing network environment. The method based on FIR neural networks can adaptively predict the traffic by learning the relationship between the past and the future traffic variations. Therefore it can be incorporated into traffic control functions in order to achieve better network performance. In [Fan 96], we propose a feedback flow control mechanism based on traffic prediction by FIR neural networks. The predicted traffic patterns in conjunction with the current queue information of the buffer can be used as a measure of congestion. When the congestion level is reached, a feedback signal is sent to sources to reduce their bit rates. Simulation results show that our scheme leads to a much lower cell loss rate than the conventional

feedback control method and hence provides a simple and efficient traffic management for ATM networks.

References

- [Amenyo 91] Amenyo, J. T., Lazar, A. A. and Pacifici, G. (1991) Cooperative distributed scheduling for ATS-based broadband networks. CTR Technical Report, Columbia University, New York.
- [Daigle 86] Daigle, J. N. and Langford, J. D. (1986) Models for analysis of packet voice communications systems. *IEEE J. Selected Areas in Comm.*, **SAC-4**, 847-855.
- [Fan 96] Fan, Z. and Mars, P. (1996) Access flow control for ATM networks using a neural network traffic predictor. to appear in *Proc. 13th IEE Teletraffic Symposium*, Glasgow, UK.
- [Funahashi 89] Funahashi, K. (1989) On the approximate realization of continuous mappings by neural networks. *Neural Networks*, **2**, 183-192.
- [Habib 92] Habib, I. W. and Saadawi, T. N. (1992) Multimedia traffic characteristics in broadband networks. *IEEE Communications Mag.*, 48-54.
- [Haykin 94] Haykin, S. (1994) *Neural Networks*. Macmillan, New York.
- [Heffes 86] Heffes, H. and Lucantoni, D. M. (1986) Markov modulated characterization of packetized voice and data traffic and related statistical multiplexer performance. *IEEE J. Selected Areas in Comm.*, **SAC-4**, 856-868.
- [Hiramatsu 90] Hiramatsu, A. (1990) ATM communications network control by neural networks. *IEEE Trans. Neural Networks*, **1**, 122-130.
- [Hornik 89] Hornik, K., Stinchcombe, M and White, H. (1989) Multilayer feedforward networks are universal approximators. *Neural Networks*, **2**, 359-366.
- [Lang 88] Lang, K. J. and Hinton, G. E. (1988) The development of the time-delay neural network architecture for speech recognition. Technical Report CMU-CS-88-152, Carnegie-Mellon University, PA.
- [Lapedes 87] Lapedes, A. and Farber, R. (1987) Nonlinear signal processing using neural networks: Prediction and system modeling. Technical Report LA-UR-87-2662, Los Alamos National Laboratory, NM.
- [Leland 93] Leland, W. E., Taqqu, M. S., Willinger, W. and Wilson, D. V. (1993) On the self-similar nature of Ethernet traffic. Bellcore Technical Report, NJ.

- [Maglaris 88] Maglaris, B., Anastassiou, D., Sen, P., Karlsson G. and Robbins, J. D. (1988) Performance models of statistical multiplexing in packet video communications. *IEEE Trans. Commun.*, **36**, 834-843.
- [Morris 94] Morris, R. J. T. and Samadi, B. (1994) Neural network control of communications systems. *IEEE Trans. Neural Networks*, **5**, 639-650.
- [Onvural 94] Onvural R. O. (1994) *Asynchronous Transfer Mode Networks*, Artech House, Boston.
- [Rasband 90] Rasband, S. N. (1990) *Chaotic Dynamics of Nonlinear Systems*, Wiley, New York.
- [Sriram 86] Sriram, K. and Whitt, W. (1986) Characterizing superposition arrival processes in packet multiplexers for voice and data. *IEEE J. Selected Areas in Comm.*, **SAC-4**, 833-846.
- [Tarraf 94] Tarraf, A. A., Habib, I. W. and Saadawi, T. N. (1994) A novel neural network traffic enforcement mechanism for ATM networks. *IEEE J. Selected Areas in Comm.*, **SAC-12**, 1088-1095.
- [Wan 94] Wan, E. A. (1994) Time series prediction by using a connectionist network with internal delay lines. in *Time Series Prediction*(eds. A. S. Weigend and N. A. Gershenfeld), 195-217, Addison-Wesley.

Biography

Zhong Fan received the BSc and MPhil degrees in electronic engineering from Tsinghua University, Beijing, in 1992 and 1994, respectively. He is now working toward his PhD at University of Durham, UK. His research interests include neural networks, traffic modeling and control of ATM networks.

Philip Mars is Professor of Electronics and Director of the Center for Telecommunication Networks at the University of Durham, UK. He is the coauthor of two research monographs and over 120 published papers. His research interests are in the application of nonsymbolic AI to telecommunications and in network performance modelling and simulation.

Analysis, simulation and experimental verification of the throughput of GCRA based UPC functions for CBR streams

F. W. Hoeksema

*University of Twente, Tele Informatics & Open Systems Group
P.O. Box 217, 7500 AE Enschede, The Netherlands
Tel.: +31 53 489 27 70, Email: hoeksema@cs.utwente.nl*

J. Kroeze

*Ericsson Telecommunication
P.B. 8, 5120 AA Rijen, The Netherlands
Tel.: +31 161 242 466, Email: etmjohk@etm.ericsson.se*

J. Witters

*Alcatel Bell
Francis Wellensplein 1, B-2018 Antwerp, Belgium
Tel.: +32 3 240 79 27, Email: jwit@rc.bel.alcatel.be*

Abstract

This paper investigates a Generic Cell Rate Algorithm (GCRA) based Usage Parameter Control (UPC) function implementing a discard function for non-conforming cells, thereby establishing a so-called Cell Discard Ratio (CDR) as UPC performance measure. Focusing on Constant Bit Rate (CBR) connections, the UPC transfer characteristics are studied in case of Peak Cell Rate Contract Violations. Also the influence of the Contracted Cell Delay Variation Tolerance value on the CDR performance of the UPC is incorporated in the study.

Algorithmic formulas found by analysis as well as simulation results are compared against the outcome of test-bed measurements. As opposed to the approach in the analysis, the simulation effectively takes into account the delay experienced by the cells accessing the ATM slotted medium, as sole origin of cell delay variation. The applicability of the simulation and the

analysis is verified with measurements on real ATM cell streams, using the R2061 EXPLOIT test-bed. Although both the simulation and analysis show clear correspondence with the measurement results, slight deviations are found. These deviations can be partly explained by the slotted nature of ATM networks, which shows the importance of taking this effect into account in the performance analysis of UPC behaviour.

The study results in guide-lines for setting the parameters involved in policing ATM CBR cell streams. These guide-lines are verified by a test-bed measurement with real CBR video data transported over ATM using AAL1.

Keywords

B-ISDN, ATM, UPC, GCRA, CBR, throughput analysis, simulation, measurements

1. INTRODUCTION

The UPC function is an ATM layer traffic control function and is located at the Public UNI (Pu-UNI) of an ATM network [6]. Its objective is to monitor and control traffic per Virtual Channel Connection or Virtual Path Connection (VCC/VPC) in terms of traffic offered and in terms of validity of the ATM connection. In the sequel the validity of a VCC/VPC is assumed.

Here we focus on policing (popular for "UPC Action") of the Peak Cell Rate (PCR or R_p) of CBR sources. This Peak Cell Rate is defined as the reciprocal of the minimal interarrival time between two consecutive requests to send an ATM_PDU (the 53 byte ATM cell) at the PHY_SAP in ATM Terminal Equipment (TE). The minimal interarrival time is called Peak Emission Interval (PEI or T_p), so: $R_p = 1/T_p$. The arrival times of the CBR input traffic are given by:

$$\{ta_{PHY_SAP@TE}[k] = ta_{PHY_SAP@TE}[1] + (k-1) T_p; k \geq 1\} \quad (1)$$

in which $ta[k]$ is the arrival time of the k -th cell of the connection.

As a result of negotiations between TE and network during the connection setup-phase the part of the Traffic Contract necessary for Peak Cell Rate policing is agreed upon. This part consists of a Contracted Peak Cell Rate R_c ($= 1/T_c$) and a Contracted Cell Delay Variance (CDV) Tolerance τ_c . The CDV Tolerance allows for a certain degree of cell clumping, and can be seen as a measure of burstiness of the cell stream at the Pu-UNI. In order to be able to specify unambiguously in the Traffic Contract which cells of a connection are conforming and which cells are not, the Generic Cell Rate Algorithm, as described by the ATM Forum [7], is used as a Conformance Definition. A Conformance Definition can be considered a deterministic means of classifying stochastic source traffic patterns.

The GCRA may not only be used to classify traffic patterns, but also as an algorithm to monitor and control traffic. The UPC function investigated in this article is GCRA(T_c, τ_c). Other algorithms which may be used as a UPC function (e.g. moving window, sliding window) have been compared against GCRA in [5]. It is assumed that the UPC function does not buffer more than one cell.

Each cell in the ATM connection is labelled conforming or non-conforming by the UPC function. Cells which are labelled non-conforming are assumed to be discarded in this work (this is not a necessity however, see [6],[7]).

So, if X is the number of cells arriving at the ingress of the UPC function since the beginning of the connection, and Y is the number of (conforming) cells at the egress of UPC function since the beginning of the connection, the number of discarded cells is $X - Y$ and we may define a Cell Discard Ratio (CDR) as

$$\text{CDR}[X] = \frac{(X - Y)}{X}.$$

Note that the CDR depends on the number of transmitted cells. The relation with the elapsed time t since the beginning of the connection can be made explicit by defining $X(t)$ and $Y(t)$ and thus

$$\text{CDR}(t) = \frac{X(t) - Y(t)}{X(t)}.$$

As the interest is in the long term behaviour of a UPC function we define

$$\text{CDR}_\infty = \lim_{X \rightarrow \infty} \text{CDR}[X]$$

or

$$\text{CDR}_\infty = \lim_{t \rightarrow \infty} \text{CDR}(t)$$

(2)

provided that these limits exist.

In the CBR case we have $X(t) = \lfloor t / T_p \rfloor + 1$ and $\lim_{t \rightarrow \infty} \frac{X(t)}{t} = R_p$.

Defining the Passed Cell Rate $R_o = \lim_{t \rightarrow \infty} \frac{Y(t)}{t}$ (provided this limit exists), we find

$$\text{CDR}_\infty = \frac{R_p \cdot t - R_o \cdot t}{R_p \cdot t} = \frac{R_p - R_o}{R_p}.$$

(3)

If a stream is not conforming to GCRA(T_c, τ_c), e.g. because $R_p > R_c$ (or equivalently because $T_p < T_c$) cells are discarded: the Traffic Contract is violated. The discarding of excess traffic is in accordance with the Traffic Contract and does not contribute to the network Performance degradation allocated to the UPC function [section 3.2.3.2, 6].

In this paper the throughput behaviour of a UPC function is considered "ideal" if the amount of discarded cells is proportional to the amount of contract violation (a property called Throughput Fairness (TF) [3],[4]), and if no cells are lost if $T_p \geq T_c$. So, the "ideal" throughput behaviour is given by

$$\text{CDR}_\infty = 1 - \frac{T_p}{T_c} \text{ for } \delta \leq T_p < T_c \text{ and } \text{CDR}_\infty = 0 \text{ for } T_p \geq T_c.$$

(4)

Restated otherwise: the "ideal" throughput behaviour of the UPC function is defined as the property to always admit the Contracted PCR, irrespective of the magnitude of the Contract Violation.

In figure 1 two views of the "ideal" throughput behaviour of a UPC algorithm are depicted.

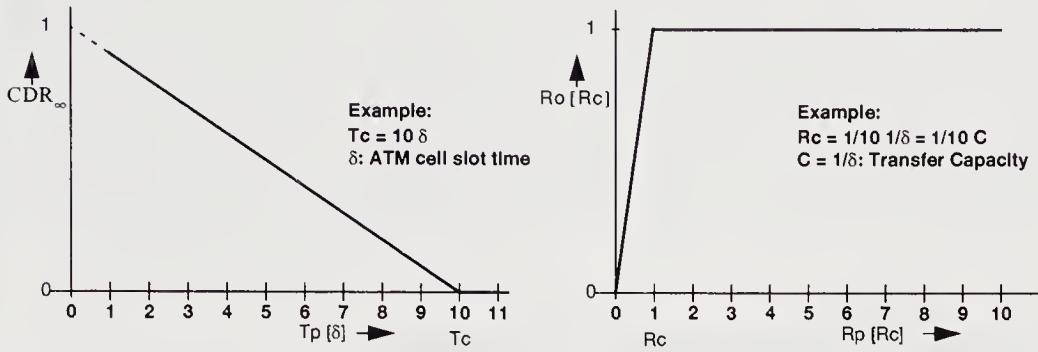


Figure 1. "Ideal" throughput behaviour of a UPC function.

The left side is a graphical representation of formula (4), while the right side of the figure shows the normalised Passed Cell Rate as a function of the Peak Cell Rate (both normalized with respect to the Contracted Cell Rate). The figure can be easily derived rewriting formulas (3) and (4) as

$$\frac{R_o}{R_c} = (1 - CDR_{\infty}) \cdot \frac{R_p}{R_c}. \quad (5)$$

δ denotes the ATM Cell Slot Time, the inverse of the Cell Transfer Capacity C [cell/s] of the link at the PHY_SAP (e.g. $C = 155.52 \cdot 10^6 / (53 \cdot 8)$ cell/s for STM-1). It is assumed that $R_p \leq C$, or equivalently $T_p \geq \delta$.

Note that our UPC function acts instantaneously on a per-cell basis, there is no shaping included (which might be beneficial from the user's point of view, provided that introduction of delay is acceptable for the service).

In this article we compare the throughput behaviour of the GCRA based UPC function with respect to Throughput Fairness, using results from analysis, simulation and test-bed measurements. Guidelines for the selection of the Contracted CDV Tolerance will be presented.

In the next section the terminal configuration, the source traffic and causes of CDV are presented. In section 3 the results of analysis and guidelines for the selection of the Contracted CDV Tolerance are given. The simulation results are presented in section 4 and test-bed measurements in section 5. In section 6 the guidelines are verified by policing a CBR video stream.

In section 7 results of analysis, simulation and test-bed measurement are compared. Section 8 contains our conclusions.

2. TERMINAL CONFIGURATION, SOURCE TRAFFIC AND CDV

A CBR source may be connected to ATM Terminal Equipment (TE) via AAL1 at a VCC/VPC endpoint. This source produces a stream of requests to send an ATM_SDU (ATM cell payload, 48 byte; interaction primitive ATM_SDU_Data.request). It is assumed that the previously mentioned stream (the user data component [section 2.3.3, 6] of the *tagged connection*) is multiplexed at the ATM layer in the TE with Operations, Administration and Maintenance (OAM) cells and cell streams of other connections. After multiplexing, the cells of the tagged connection are considered to be shaped by a shaper, resulting in a stream with PEI T_p .

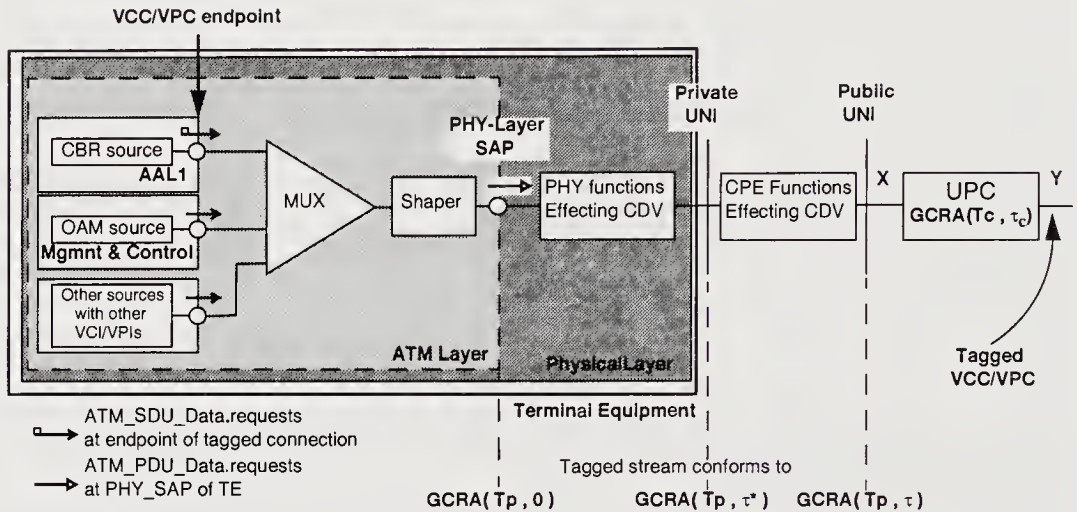


Figure 2. Terminal Example.

In figure 2 a terminal example is given, using an adaptation of both figure 4 of I.371 [6] and the PCR Reference model of the ATMF UNI 3.0 specification [7].

At the PHY_SAP of TE the cell stream (stream of ATM_PDU_Data.requests) of the tagged connection has an PEI of T_p , a result of the shaper. However, this is not the case any more at the Public UNI (stream of ATM_PDU_Data.indications at the PHY_SAP of Pu-UNI). Cell Delay Variation is introduced by the following mechanisms:

- 1/. Due to the *ATM multiplexing* with OAM cells and cells of other connections at the PHY_SAP in TE some cells may be delayed.
- 2/. The ATM_PDU_Data.indications at the PHY_SAP of the Pr-UNI only occur at discrete time instances. This so called *slotted nature* of ATM networks causes CDV.
- 3/. Due to insertion of *PHY layer overhead* the ATM cell slot times may not be of equal length.
- 4/. Between the Pr-UNI and Pu-UNI CDV may be introduced by *Customer Premises Equipment* (CPE).

Note that the precise nature of CDV causes in the CPE is left unspecified here. It *may* consist of, but not be limited to: ATM multiplexing, the slotted nature of ATM networks or PHY layer overhead.

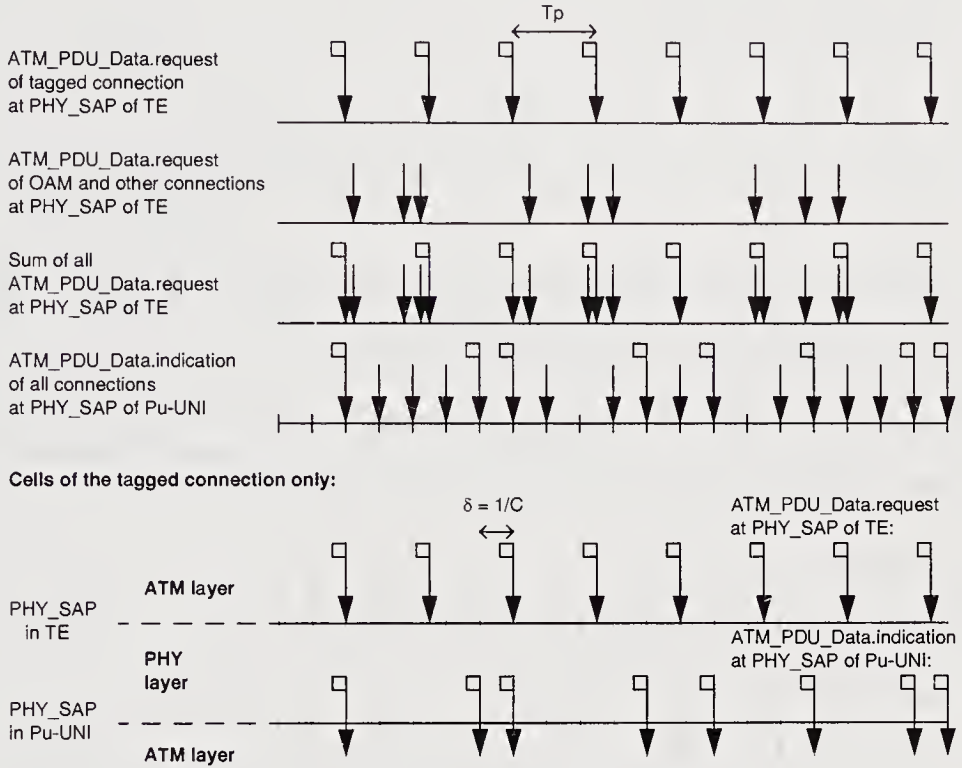


Figure 3. CDV of cells of the tagged connection (no CPE caused CDV).

In Figure 3 an example of the introduction of CDV is given (CDV cause 1/. to 3/., no CDV caused by 4/.). Note that the shaper guarantees a PEI of T_p at the PHY_SAP of TE for the tagged connection. Competing ATM_PDU_Data.requests at the PHY_SAP in TE are assumed to be in continuous time (see the summation in the figure above).

In the following analysis (section 3) all these four causes of CDV are neglected. In the simulation (section 4) however, the slottedness (cause 2/.) is taken into account. During the measurements with an ATM traffic generator CDV causes 3/. and 4/. may be present (section 5). Finally, the traffic from a TV Terminal Adapter may experience all CDV causes mentioned above (see section 6).

3. ANALYSIS

In figure 4 the Cell Streams of figure 2 are given, only showing CDV caused by the slottedness of ATM networks (CDV cause 2/.)

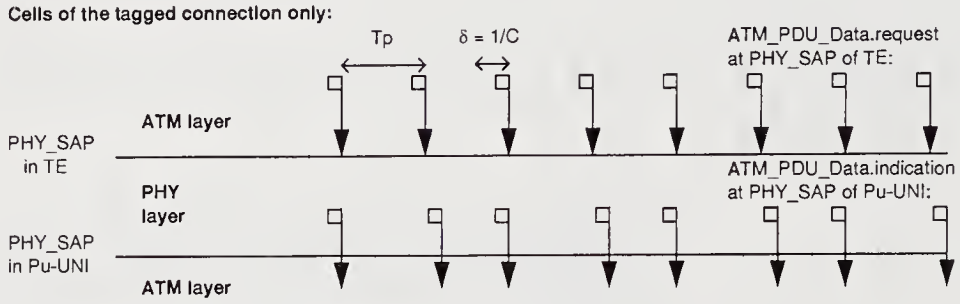


Figure 4. CDV caused by the slotted nature of ATM only.

The slotted nature of the ATM networks is neglected in the presented expressions but is important in comparing our results with results of simulation experiments [3],[4] and test-bed measurements [12]. The assumption we make is that (apart from a constant delay):

$$ta_{PHY_SAP@Pu-UNI}[k] = ta_{PHY_SAP@TE}[k] \quad (6)$$

while, when 2/. is taken into account (see figure 4):

$$ta_{PHY_SAP@Pu-UNI}[k] = \lceil ta_{PHY_SAP@TE}[k] / \delta \rceil \delta \quad (7)$$

Only the situation of Traffic Violation is investigated, thus $\delta < T_p < T_c$, no cells are lost if $T_p \geq T_c$. In the sequel we will present CDR_∞ as a function of T_c , T_p and τ_c ; a derivation of these results is given in [13].

Three cases can be distinguished, depending on the Contracted CDV τ_c :

- $\tau_c = 0$

In this case:

$$CDR_\infty = 1 - 1/N, \text{ with } N = \lceil T_c / T_p \rceil \quad (8)$$

- $0 < \tau_c < T_p$

In this case an *algorithmic solution* was found in [13]. It consists of finding P and M (both required to be integer and as small as possible) which satisfy:

$$M T_c / T_p + 1 \leq P < M T_c / T_p + 1 + (1 - \tau_c / T_p) \quad (9)$$

or equivalently:

$$P < (M T_c - \tau_c) / T_p + 2 \leq P + (1 - \tau_c / T_p) \quad (10)$$

As $P > M$ the algorithm should start with $M = 1$.
Then:

$$CDR_{\infty} = 1 - M / (P-1) \quad (11)$$

$$\bullet \tau_c \geq T_p$$

It can be shown that the throughput behaviour is "ideal" in this case, so:

$$CDR_{\infty} = 1 - \frac{T_p}{T_c} \quad (12)$$

Now, we will present some results of the analysis which can be directly compared to the results of simulation (section 4) and test-bed measurements (section 5). Comparisons will be deferred until the simulation and test-bed results are presented.

In figure 5 the CDR as a function of the PEI T_p is presented for $T_c = 10 \delta$ for different values of the Contracted CDV Tolerance τ_c . The "ideal" throughput curve is shown too (see figure 1).

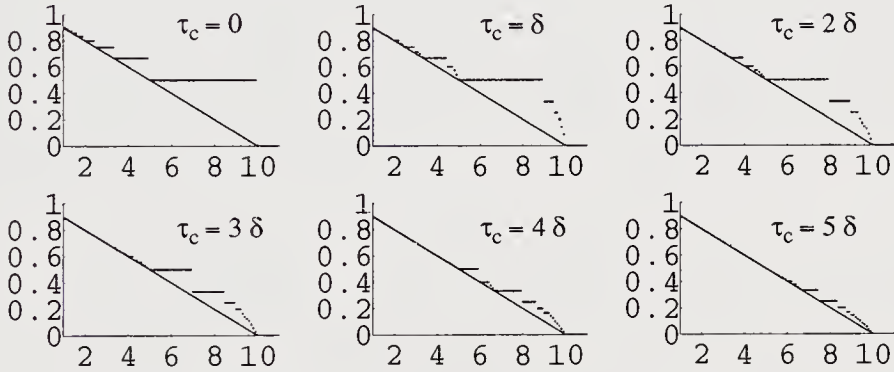


Figure 5. CDR_{∞} as a function of $T_p [\delta]$ for $T_c = 10 \delta$. Different values of τ_c .

Figure 6 shows the Passed Cell Rate R_o as a function of the Peak Cell Rate R_p for both the "ideal" behaviour and the behaviour for $\tau_c = 8 \delta$ with $T_c = 63.75 \delta$.

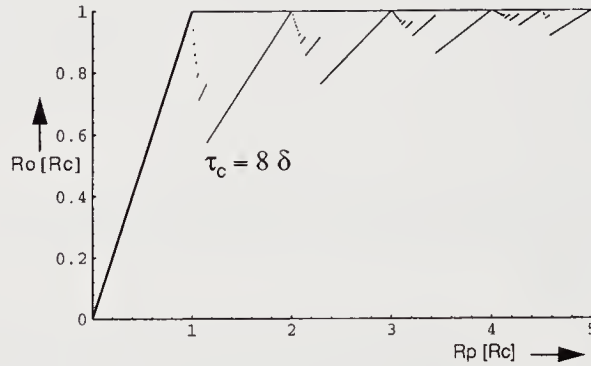


Figure 6. Passed Cell Rate R_o as a function of PCR.

Figure 7 offers a closer look at the previous figure, and shows the influence of a change in Contracted CDV Tolerance.

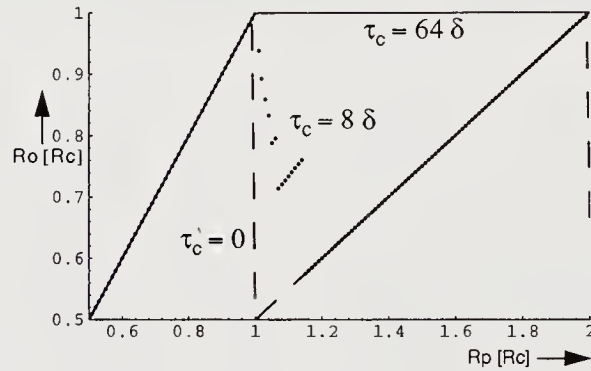


Figure 7. Passed Cell Rate R_o as a function of PCR.

From our analysis it is clear that the throughput behaviour for $\tau_c = 0$ is far from "ideal". Increasing τ_c in the region $0 < \tau_c < T_p$ improves the throughput behaviour of the GCRA based UPC function, but does not realize "ideal" throughput behaviour either. Only when $\tau_c \geq T_p$ the situation is reached in which the Throughput Fairness property holds.

These observations allow us to provide guidelines for the selection of the Contracted CDV Tolerance if Throughput Fairness is required:

In the connection set-up phase, the user presents his requested PEI $T_u = T_p$ and requested CDV Tolerance $\tau_u = 0$ to the network. For the UPC function to perform "ideal" throughput behaviour our analysis shows that the contracted values should be $T_c = T_p$ and $\tau_c = T_p = T_c$. If a user intends to violate the Traffic Contract and specifies the required PEI as $T_u = T_p' > T_p$ (the actual PEI) and required CDV Tolerance as $\tau_u = 0$, the network will select $T_c = T_p'$ and

$\tau_c = T_p' = T_c$ if "ideal" UPC throughput behaviour is required. From our analysis we notice that the Contracted CDV Tolerance is too high in this case, it could be $\tau_c = T_p < T_p'$. As the network has no idea of the intentions of the user the only "fair" selection of $\tau_c = T_c = T_p'$.

In [3] it is shown that the following approximation of the CDR holds when Δ is fairly small and $T_c \geq \tau_c$:

$$CDR_{\infty} \cong T_c * \Delta / \tau_c, \text{ where } \Delta = (T_c - T_p) / T_c \quad (13)$$

a result which is in accordance with [1], [4] and [2], where a more general relation is given,

$$CDR_{\infty} = 1 - d / (d + 1), \text{ with } d = \lceil \tau_c / (T_c - T_p) \rceil \quad (14)$$

which is in agreement with the algorithm in (8), (9) and (10). However, note that (14) only holds for $0 < \tau_c < T_p$ and $T_c/2 \leq T_p < T_c$ as is stated as a (strong) conjecture in [13]. Using (14) it can be shown that the approximation in (13) is in fact an upper bound to CDR_{∞} , which becomes tighter as $\Delta \ll 1$. So, for small contract violations (13) shows that for $\tau_c \geq T_c$ indeed $CDR_{\infty} \cong \Delta$.

4. SIMULATION

The GCRA based UPC function is analysed by simulation techniques taking into account the delay experienced by cells accessing the ATM slotted medium (CDV cause 2/., section 2). ATM cells are generated periodically by a CBR source (on the real time axis). Due to the transfer to the ATM slotted medium, cells can be delayed while others are not influenced, thus introducing a limited CDV (see figure 4). This cell flow is then passed to the UPC function where the GCRA is executed and the CDR is measured.

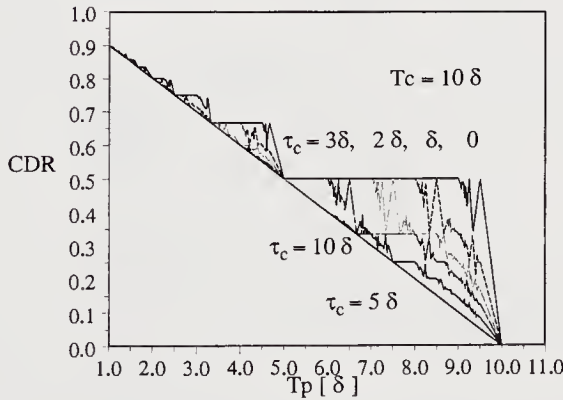


Figure 8. CDR as a function of T_p . $T_c = 10 \delta$.

The CDR in case of a Contracted PEI $T_c = 10 \delta$ is shown in figure 8 as a function of the actual PEI and of the Contracted CDV tolerance τ_c . Comparing the results with the analytical ones (see figure 5), there are a lot of similarities. The major differences are the oscillations marking the transitions between two regions of constant CDR which occur at PEIs which are *not* an integer

multiple of the cell slot time δ . In that case some cells experience a delay due to the access to the ATM slotted medium while others are not delayed, so the arrivals at the UPC function are not strictly periodical any more (e.g. figure 4, lower part)

Indeed, a CBR connection which exceeds the Contracted PCR R_c by only a small amount immediately experiences a severe Cell Discard Ratio, disproportionate with the degree of misbehaviour Δ (see (13)). On the other hand a source which sends twice the amount of allowed traffic sees a CDR of 50%, a value in accordance with the TF property.

From the simulations it became clear that a GCRA based UPC function tuned with the Contracted PEI T_c and $\tau_c = \lceil T_c / \delta \rceil \delta \geq T_c$ obtains Throughput Fairness. Similar findings have been reported in [8] when observing the UPC responsiveness characteristics.

5. MEASUREMENTS

The initial aim of the experiments was to validate the correct operation of the implemented UPC function, see [12]. Here the measurement results are used to verify the applicability of the analysis and simulations in the previous sections, using a real ATM network.

The experiments were performed at the EXPLOIT test-bed, which has four ATM switches, several terminals with appropriate adapters and measurement equipment (see [10]). The Police Function Board (PFB) can implement several UPC mechanisms, like Leaky Bucket (LB), jumping window, moving window etc. Here we use the LB mechanism as UPC function. The PFB is part of one module in the Remote Unit (RU) which is one of the available switches at the test-bed.

Figure 9 shows a simplified view of the experiment configuration. As indicated by the arrows,

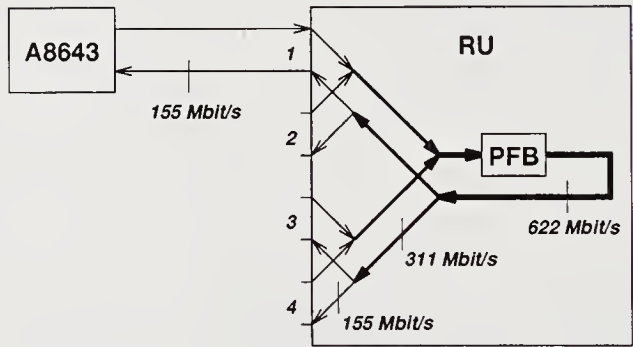


Figure 9. EXPLOIT Experimental configuration.
RU: Remote Unit (a switch); PFB: Police Function Board.

the traffic from four input ports is multiplexed to one stream entering the PFB. Cells may therefore always experience delay variations within the RU before being policed. Even with only one source connected, CDV is caused by Operations Administration and Maintenance (OAM) traffic inside the RU (interpreting the part of the RU before the PFB as CPE allows us to model this as CDV cause 4/). Traffic leaving the PFB is routed back through the RU and demultiplexed to one of the four output ports of this module.

The ATM traffic streams were generated, received and analysed with the Alcatel 8643, a PC-card with memory for 8192 assigned ATM cells. This memory can be played out repetitively, while counters keep track of the number of sent and received cells.

The implemented LB is a discrete state realisation of the GCRA and is defined by three discrete parameters: *splash*, *leak rate* and *bucket limit*. The *bucket level* is also discrete and ranges from 0 to 65535 units. With every passed cell, a splash (from 0 to 255 units) is added to the bucket level, which leaks with a constant rate, selectable by factors of two from 2^{-14} to 2^7 units/slot. A cell arriving at the PFB is discarded if the bucket level exceeds the bucket limit, which can be set from 0 to 65504 in steps of 32. These calculations are all performed by the Police Criterion Chip (PCC) [11].

The PFB parameters are related to the GCRA parameters by:

$$T_c = \text{splash} / \text{leak rate} \qquad \tau_c = \text{bucket limit} / \text{leak rate} \qquad (15)$$

With appropriate parameters, the PCC can act as a cell counter such that the number of received and discarded cells at the PFB are known. This allows to verify that all generated cells make it to the PFB and that cells are only lost due to discards and not by e.g. buffer overflow. Instead of the PFB parameters we will in the following use the parameters T_c and τ_c , both expressed in slots at 155.52 Mbit/s.

Some initial measurements where the PCR R_p and Contracted PCR R_c were exactly the same ($T_p = T_c = 64 \delta$) revealed a Cell Discard Ratio of approximately $2 \cdot 10^{-6}$, although τ_c was chosen large enough to allow all possible CDV. These discards are due to frequency deviations between the crystal of the free-running clock of the A8643 and the clock crystal of the RU, to which the PFB is synchronised. Note that, although this causes cell discards, it is *not* one of the CDV causes mentioned in section 2. To prevent these discards, a slightly lower value of $T_c = 63.75 \delta$ is used for further measurements.

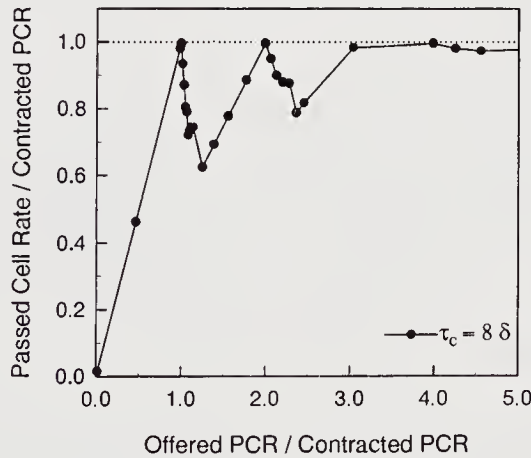


Figure 10. Passed Cell Rate R_o as a function of PCR.

Figure 10 depicts the measured throughput of the LB as a function of the PEI T_p of the A8643 for a Contracted PEI T_c of 63.75 slots (at 155.52 Mbit/s). The Contracted CDV Tolerance is $\tau_c = 8 \delta$. Mind that the dots in figures 10 and 11 are the *only* measurement points, the line pieces just connect the dots in the correct order. The generated cell streams of the A8643 only have T_p values which are integer multiples of the cell slot time, so that the cell inter-departure times at the egress of A8643 are strictly constant, e.g. the *pattern* as shown in the upper part of figure 4.

However, on the way to the RU and the PFB, the cell stream probably suffers CDV by multiplexing with OAM cells in the RU (CDV cause 4/. in section 2). ATM multiplexing in TE (cause 1/.) does not play a role. Due to the integer values of T_p the slottedness of ATM networks (cause 2/.) is not taken into account. Physical layer overhead (cause 3/.) may play a role.

The throughput is defined as the ratio of the actual cell rate of the A8643 and the Contracted PCR, multiplied by the ratio of the numbers of passed and sent cells (so R_o/R_c , see (5)). Both axes are normalised to the Contracted PCR. This figure can be directly compared to figure 6.

It is clear from figure 10 that the throughput depends both on the offered PCR and on the CDV tolerance. As long as the actual PCR does not exceed the Contracted PCR, all cells pass the LB. If however the actual PCR increases above the contracted value, the passed traffic is limited to the Contracted PCR, even if the A8643 generates at full link rate. Two measurements with a PCR slightly below and above the contracted one can not be distinguished in the plot, but the results are as expected. We see again that the punishments imposed by the LB are not proportional to the violation. This is shown more clearly in figure 11, which covers the region of small contract violations (compare with figure 7). The dips in the throughput curve vanish when the CDV tolerance is increased to the Contracted PEI.

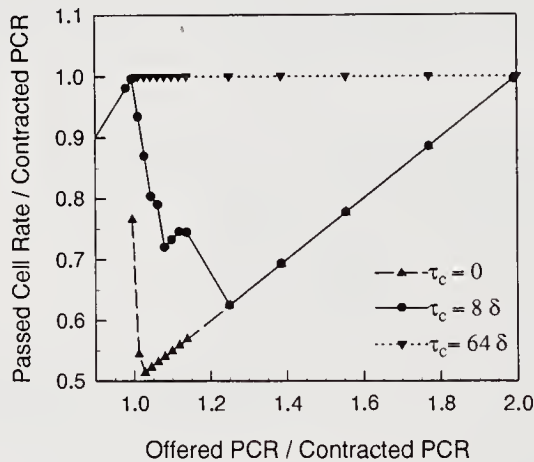


Figure 11. Passed Cell Rate R_o as a function of PCR.

6. PEAK CELL RATE CONTROL FOR DIGITAL VIDEO

The experience gained in policing CBR sources was used to police one of the real sources available at the EXPLOIT testbed, namely the TV signal (audio and video signal) which originates from the TV-Terminal Adapter (TV-TA). The (composite) TV signal is converted to a digital signal with a bit rate of 34.368 Mbit/s for the video signal and 0.96 Mbit/s for the audio signal. These "reference" signals will eventually be packetized into ATM cells according to the AAL1 adaptation layer standard for CBR traffic. Before that however, Forward Error Correction (FEC) combined with bit or byte interleaving is used for error correction [9].

For the video component a Reed-Solomon code is used which allows for an error correction of at most 4 cells out of 64 consecutive cells. So the source characteristics undergo changes because of the introduction of FEC overhead and (in AAL1) signal timing recovery. This results into a ATM physical rate of 41.366 Mbit/s for the video signal and of 1.49 Mbit/s for the audio component.

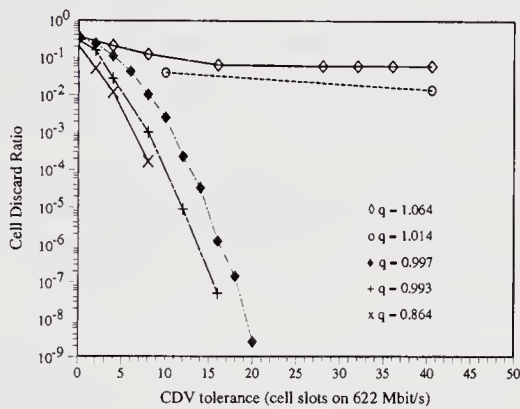


Figure 12. Measured CDR as a function of Contracted CDV Tolerance. $q = R_p / R_c$

In the policing experiments, the TV-set is continuously sending traffic which is passed through the RU (where the PFB is located) to a TV screen. The Contracted PCR (the PCR which is used for policing the source) and the Contracted CDV tolerance are changed for each experiment and the Cell Discard at the PFB is measured. In figure 12 the CDR is plotted as a function of the Contracted CDV tolerance τ_c and for different PCRs of the video source which are expressed in terms of the Contracted PCR R_c ($q = R_p/R_c$, $R_p = 41.366$ Mb/s). The Contracted PEIs T_c can be derived from the q -values by:

$$T_c = q T_p = q C / R_p \delta. \quad (16)$$

This results in $T_c = 16.0 \delta$, 15.2δ , 15.0δ , 14.9δ and 13.0δ for the q -values in figure 12.

If the actual PCR of the source is higher than the Contracted PCR (the theoretically Contract Violation region, defined by $q > 1$), then changing the CDV tolerance (within reasonable limits) has only a very small influence on the CDR (this can be seen in the upper right part of figure 12). Once the CDV tolerance is larger than the inverse of the Contracted PCR ($= T_c$), the UPC

discards exactly the excess amount of traffic (this result was found also earlier, see figure 11, in which the passed cell rate R_o equals the Contracted PCR R_c , provided τ_c is large enough).

This observation leads to a method which allows us to derive experimentally the actual PCR of a policed CBR source, once the CDR is measured. In case of a Contract Violation and a CDV Tolerance value larger than the Contracted PCR, we observed that $R_o = R_c$, so using (5) we find:

$$R_p = R_c \cdot \frac{1}{1 - \text{CDR}} \quad (17)$$

Indeed, if this method is applied to the experimental results, we find a PCR of 41.3666 Mbit/s on the 155.52 Mbit/s link and so a "reference" signal rate of 34.368 Mbit/s exactly as mentioned earlier. The same procedure was successfully used to determine the PCR of the audio component of the TV signal.

If the actual PCR of the source is below the Contracted PCR (the theoretically no Contract Violation region, defined by $q < 1$, the lower left part of figure 12), all cells should be accepted if the assumptions used for the analysis hold. From these measurement results we conclude that at least one of the mentioned CDV causes (section 2) is present. The clock mismatch problem (section 5) is believed not to influence these results, since the clocks of the TV-TA and PFB were synchronised.

Since apparently, in realistic situations, cells always experience some CDV, increasing the CDV tolerance seems necessary. It drastically reduces the CDR, even if no discards are expected (being in the theoretically no Contract Violation region, $q < 1$). This is especially true if the PCR approaches the Contracted PCR ($q \approx 1$). In order to draw further conclusions we need to model the RU more precisely.

7. COMPARISON OF ANALYSIS, SIMULATION AND MEASUREMENTS

When comparing the results of analysis and simulations, the impact of the slottedness of ATM networks on the throughput behaviour of a GCRA based UPC function becomes clear. Mind that the CDR values at integer values of T_p (see figure 8) should equal the values of CDR_∞ at integer values of T_p in figure 5, something which graphically appears to be the case (and shows that the number of simulated arrivals was large enough).

Comparing the results of the test-bed measurements with those of the analysis shows a good agreement between the two (at least graphically). From this we conclude that the effects on discards caused by the clock-mismatch between A8643 and PFB are effectively undone by a slight adaptation of the Contracted PEI T_c . Note that the impact of the slottedness of ATM was not encountered in these experiments, as the cell arrival process had only PEIs at a multiple of the cell slot time. Apparently, the OAM traffic generated by the RU (CDV cause 4/, section 2) does not alter the measured throughput function significantly.

The CBR video experiment (section 6) showed that increasing the Contracted CDV Tolerance, even in the theoretically no Contract Violation case is beneficial from a UPC discard point of view. This was not found using the other approaches, which shows the importance of

the measurements. However, the necessity of detailed modelling of the system on which the measurements were performed is clear.

8. CONCLUSIONS

In this paper the throughput behaviour of a GCRA based UPC function for ATM traffic control has been studied by analysis, simulation and measurements.

The analysis resulted in an algorithm to compute the Cell Discard Ratio CDR_{∞} for infinite length CBR traffic streams. The CDR is a function of the contracted GCRA parameters T_c , τ_c and the PEI T_p of the cell stream. The analysis does not take any CDV generating mechanism into account, and assumes that cells arrive in continuous time, spaced T_p apart, at the ingress of the UPC function.

The results of the simulation are in-line with the analysis results as far as comparable. The simulation takes the CDV due to the slottedness of the physical layer into account. The influence of the slottedness on the throughput behaviour of the UPC function is limited but significant. No results of analysis have been found by the authors which take the slottedness of ATM into account, although the methods presented in [13] may prove valuable in solving this issue.

Measurements with an ATM traffic generator and an implemented UPC function show that the analysis and simulation apply to real ATM networks. Analysis and simulation show remarkable resemblance with the measurement results. Effects of clock-mismatch can be undone effectively by a slight adaptation of T_c , at least in the case where no CDV due to slottedness is present.

Measurements with real ATM audio and video traffic however, show that apart from the slottedness of ATM networks, other CDV causes have their influence on the throughput of the UPC function as well. This is especially apparent if the PCR approaches the Contracted PCR. In this case increasing τ_c drastically reduces the cell discard. In order to explain these observations, the experimental configuration has to be modelled in greater detail.

Guidelines for the selection of T_c and τ_c were presented, based on the results from analysis and simulation. These guidelines proved useful when applied to a realistic situation, although also in the theoretically no Contract Violation case, τ_c has to be increased. As to how much exactly, guidelines still need to be found. This is an item for further study.

The research may have a follow-up by extending analysis and simulation by inclusion of more CDV causes. The two approaches should be tested with measurements in systems that are described at a level of detail required by the models used.

A study consisting of analysis, simulation and measurements of the policing mechanism for traffic using other ATM Transfer Capabilities (a.k.a. ATM service categories or connection types) seems necessary, e.g. for Variable Bit Rate traffic, for Available Bit Rate traffic or for Signalling traffic.

Policing of the Sustainable Cell Rate is another candidate subject for further study.

9. ACKNOWLEDGEMENTS

The measurements have been carried out as part of the RACE project R2061 EXPLOIT. The authors would like to thank all the people who contributed to EXPLOIT, especially work package 3.1 who supplied the measurement results as presented in this paper and T. Renger for providing the plots.

10. REFERENCES

- [1] S. Liu, T. Chen, V.K. Samalam and J. Ormond. "Performance Analysis of GCRA for CBR Sources". ATM Forum contribution 94-0182, March 1994.
- [2] B.G. Kim and I.G. Niemegeers. "Assessment of Traffic Control Schemes in ATM Networks". Memoranda Informatica 94-46, TIOS 94-16. ISSN 0924-3755. August 29 1994.
- [3] J. Witters, S. Muti and G.H. Petit. "Performance Analysis of a VSA-based UPC Function for Constant Bit Rate Sources". Alcatel-Bell Internal Research Report, TTD_068/JW_930330, Ed. 1, 30/03/93.
- [4] R. Wilts, J. Witters and G.H. Petit. "Throughput Analysis of a UPC Function Monitoring Misbehaving CBR Sources". Proc. Second Workshop on Performance Modelling and Evaluation of ATM Networks. 4-7 July, 1994. Bradford, UK. pp. 39/1-39/17.
- [5] H. Hemmer. "Evaluation of UPC Functions for Peak Cell Rate Enforcement". European Transactions on Telecommunications, Vol. 5, nr.2. March - April 1994. pp. 27-31.
- [6] ITU-T Recommendation I.371: Traffic Control and Congestion Control in B-ISDN, Frozen Issue - Paris, March 1995.
- [7] ATM forum UNI specification version 3.0. Prentice Hall, 1993.
- [8] J. Witters, G.H. Petit & S. Muti. "Responsiveness Characteristics of a Usage Parameter Control Function based on the Virtual Scheduling Algorithm Monitoring Misbehaving Constant Bit Rate Sources". ITC Seminar on Digital Communication and Network Management, St. Petersburg, Russia, June 15-20, 1993.
- [9] H. Hessenmuller, S. Nunes. "A terminal Adapter for High-Quality Audio and Video Signals to be used in a B-ISDN based on ATD". Proc. of the International Symposium on Broadcasting Technology, Beijing, September 1991.
- [10] M. Potts. "EXPLOITation of an ATM Testbed for Broadband Experiments and Applications". Electronics & Communication Engineering Journal, December 1992, pp. 385-393.
- [11] K. van der Wal, M. Dirksen, D. Brandt. "Implementation of a Police Criterion Calculator based on the Leaky Bucket Algorithm". Proc. of the IEEE Globecom '93, Houston, Nov. 29 - Dec. 2, 1993, pp. 713- 718.

- [12] A. Bohn Nielsen, R. Elvang, H. Hemmer, H. Pettersen, J. Kroeze, T. Renger, J. Witters. "Results of experiments on traffic control using test equipment". RACE project R2061 EXPLOIT, Deliverable 18, 30.06.94.
- [13] F.W. Hoeksema. "On the performance of GCRA based UPC functions for Peak Cell Rate policing". Proc. B-ISDN Teletraffic Modelling Symposium, Antwerp, Alcatel Bell, February 17, 1995.

11. BIOGRAPHY

Fokke Hoeksema studied electrical engineering at the University of Twente. He received his MSc degree in 1987 after which he joined Philips Research. He participated in the design of a high-quality high-resolution still-image coding system and contributed to modelling of CCD image sensors.

In 1989 he joined the department of electrical engineering of the University of Twente, where he worked in the field of subband image-coding and packet video (BOAT project, together with KPN research).

Currently, as an assistant professor in the Centre of Telematics and Information Technology (CTIT), he is working on design and analysis of architectures and protocols for Broadband Telecommunication networks. He investigated ATM traffic control mechanisms and is currently working on QoS improvement methods that use the outcomes of performance measurements to close control loops in ATM based networks.

John Kroeze (born 1965) studied electrical engineering at the University of Twente, which resulted in a MSc degree in 1992, after working on his thesis at the Dutch PTT Research laboratory in Leidschendam.

He worked for three years at the Computing Science Institute of the University of Nijmegen, participating in the European RACE II project R2061 EXPLOIT, for which he was involved in ATM traffic control experiments.

Recently he has joined Ericsson Telecommunications in Rijen, the Netherlands, where he works at the Intelligent Networks Application Laboratory.

Johan Witters studied physics at the University of Antwerp (Belgium) where he obtained his Ph.D in 1992 in the area of Quantum Physics with a theoretical study of superfluid helium.

In the same year he joined the research centre of Alcatel Bell in Antwerp.

Since then, he is working on ATM traffic control. His main interest is on Usage Parameter Control both from a theoretical as well as an experimental point of view.

The theoretical aspects involve both simulations and queuing techniques. For the experimental side, he has participated in RACE project EXPLOIT.

Now he is active in the ACTS project EXPERT. Currently he works on policing for the Available Bit Rate Service.

When Is Traffic Dispersion Useful? A Study On Equivalent Capacity

E. Gustafsson

Royal Institute of Technology, Dept. of Teleinformatics

KTH Electrum/204, S-164 40 KISTA, SWEDEN

Phone +46 8 752 14 98, Fax +46 8 751 17 93, Email: evag@it.kth.se

G. Karlsson

Swedish Institute of Computer Science

Box 1263, S-164 28 KISTA, SWEDEN

Phone +46 8 752 15 77, Fax +46 8 751 72 30, Email: gk@sics.se

Abstract

Multi-media and data traffic are anticipated to occupy much of the resources in integrated services networks, based on ATM. These traffic types appear to exhibit strong autocorrelation over long periods, which affects the performance of statistical multiplexing detrimentally. The correlation has most commonly been handled by spreading the traffic in time, so called shaping, which may introduce considerable delay.

We take a different approach, namely spreading the traffic in space over multiple, independent paths. The autocorrelation in the traffic is thereby reduced and bursts are spread out. This alleviates queuing delay and, for a given quality level, lowers the capacity needed for each transmission. We denote this strategy *traffic dispersion*.

In this paper, we focus on how traffic dispersion affects the equivalent capacity needed for a transmission. By studying its behaviour, we can determine under what circumstances spatial traffic dispersion is motivated for different cost functions, when using a certain number of paths in the network. The first cost function is a fixed charge per capacity unit. Next, we add a fixed charge per connection to the previous cost, and lastly, we let the charge per path increase progressively. Our findings show that spatial traffic dispersion alleviates the most troublesome traffic cases, that is, those with a high peak-to-mean ratio and those with a high peak-to-link ratio. Furthermore, the cost benefits due to dispersion seem to justify the extra effort needed to implement it.

This work was in part presented at the IFIP TC6 Third Workshop on Performance Modelling and Evaluation of ATM Networks, Ilkley, U.K., July 1995.

Keywords

Traffic dispersion, multi-path routing, equivalent capacity, traffic control, ATM networks.

1 INTRODUCTION

The asynchronous transfer mode (ATM) is the network architecture that the International Telecommunication Union recommends for broadband integrated services digital networks. Succinctly described, the mode combines the circuit switched routing of telephone systems with the statistical multiplexing of packet switching. This is accomplished by establishing a connection (fixed route) through the network before accepting any traffic. The information is then sent over the connection in 53-octet long cells, which are routed according to address information contained in their 5-octet headers.

The capacity of a transmission link is statistically shared among the connections traversing it. When traffic arrives randomly, the capacity offered by the link occasionally becomes insufficient. This could be handled by buffering, but as the arrivals come in longer and longer bursts the buffers will eventually overflow and cells will be lost.

Earlier studies have shown that the probability of cell loss is highly dependent on the correlation in the multiplexed traffic stream, Li (1989). For a given connection, the correlation can be lowered by spreading the traffic in time, so called shaping. This method may however give rise to delays too large to be tolerated by the application. So, statistical multiplexing of traffic streams with strong correlation would, at a low probability of cell loss, require unreasonable low utilization of the network resources and excessively large buffers.

Ever since Maxemchuk's contribution, Maxemchuk (1975), there have been several different suggestions for spreading the traffic from a source in space rather than in time, as a means for load balancing and fault handling in packet-switched networks, Gustafsson (1994:1). Spatial traffic dispersion means that a message is divided into a number of sub-messages, which are transmitted in parallel over disjoint paths in the network, as shown in Figure 1. A large burst of data will consequently be sent as more moderately sized bursts, and the correlation will be reduced without the extra delay that temporal shaping would introduce.

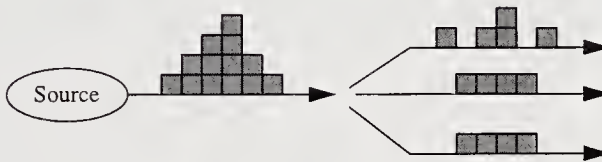


Figure 1 Illustration of spatial traffic dispersion.

The traffic from a source is transmitted in parallel through the network, and resequenced at the receiver. Dispersion should be possible in any network where disjoint paths exist between the source and the destination. Given a number of such paths, the traffic may be spread according to different strategies. One possibility is to spread the packets in the traffic stream cyclically over the paths - a solution which is discussed in Lee and Liew (1993), Maxemchuk (1993). Another way would be to submit the packets in longer sequences on each path, as suggested for the string mode protocol, Déjean et al. (1991), and yet another solution would be to spread the traffic dynamically over the paths, Cheng (1994). The latter variant would however require substantially more overhead. Essentially, a spreading strategy should apply to the traffic characteristics in order to minimize the correlation in the resulting traffic streams, since lowering the correlation is one of the main advantages of traffic dispersion.

Spatial traffic dispersion thus improves statistical multiplexing and it also enhances network security, as eavesdropping on several connections simultaneously may be difficult. Since the dispersion scheme employs disjoint paths, cell loss on one connection is independent of losses on other connections, and forward error correction can successfully be used to correct the losses. Regarding these advantages, the question arises whether traffic dispersion is useful under all circumstances.

To answer this question, or at least give a hint, we have chosen to focus on how traffic dispersion affects the equivalent capacity of a transmission. The equivalent capacity is the predicted capacity, that for certain source characteristics and demands on performance, needs to be allocated on a link. We investigate for what values of source peak rate and source mean rate, and their relation to the link capacity, spatial traffic dispersion is useful.

Equivalent capacity is discussed in Section 2, while Section 3 covers the cost functions used in the evaluations. Our results are presented in Section 4, and Section 5 concludes the paper.

2 EQUIVALENT CAPACITY

2.1 Equivalent capacity without buffering

The ATM concept makes use of statistical multiplexing which allows multiple sources to share a link statistically. This means that the demand for capacity at times may exceed the available resource on the link, and cells will be lost. Given a limit on the cell-loss probability, we can calculate the maximum number of identical sources n which can be multiplexed on a link of capacity C . The equivalent capacity required for one source is then C/n .

If the objective is to maintain the traffic arrival rate below the link capacity, that is, assuming no buffering, the cell-loss probability may be approximated by:

$$\phi = \frac{E\{(\lambda - C)^+\}}{E\{\lambda\}}, \text{ where } \lambda \text{ is the arrival rate, and } (\lambda - C)^+ \text{ is } \max\{0, \lambda - C\}. \quad (1)$$

Define the arrival process to consist of n independent identically distributed on-off sources, each with peak rate h (Figure 2). The model is chosen to get a tractable expression for the equivalent capacity while capturing some of the burstiness that can be anticipated from future traffic sources.

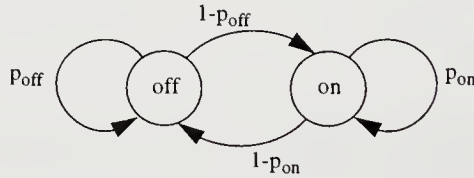


Figure 2 The state diagram of an on-off source. The system stays off with probability p_{off} , and once on, it remains on with probability p_{on} .

While in active state, the source generates traffic at peak rate h . The mean rate of the source is thus given by εh , where ε is the fraction of time that the source spends in active state:

$$\varepsilon = \frac{1 - p_{off}}{2 - p_{on} - p_{off}}. \quad (2)$$

As the number of simultaneously active sources is binomially distributed, the cell-loss probability is given by

$$\varphi = \frac{1}{n \cdot \varepsilon h} \sum_{x=\frac{C}{h}}^n (xh - C) \Pr \{x \text{ sources on}\} = \frac{1}{n \varepsilon} \sum_{x=\frac{C}{h}}^n \left(x - \frac{C}{h}\right) \binom{n}{x} \varepsilon^x (1 - \varepsilon)^{n-x}. \quad (3)$$

The burstiness of a source is in these equations defined as

$$\frac{\text{Source peak rate}}{\text{Source mean rate}} = \frac{h}{\varepsilon h} = \frac{1}{\varepsilon}, \quad (4)$$

and the cell-loss probability is hence dependent on the ratio C/h as well as on the source burstiness.

Since the correlation between cells generated by the source in Figure 2 is monotonously decreasing, cyclic dispersion would minimize the correlation on each path. This is thus the dispersion strategy preferred, and it is assumed in the remainder of this paper. A dispersed source, as the link experiences it, may be approximated by another on-off source with the same characteristics except that the peak rate is reduced to h/N , N being the dispersion factor. The dispersion factor is defined as the number of paths over which the traffic from a source is spread. In the following, a dispersed source represents the traffic that an original source sends over one of the paths. We show the effects of dispersion on the equivalent capacity by replacing each original on-off source by N independent dispersed sources.

Essentially, what we do is modelling an on-off source with peak rate h as N on-off sources, each with peak rate h/N (Figure 3). These sources would be completely correlated, since they together represent the original source. With dispersion however, the traffic from each of these N sources is sent over a separate path, disjoint from all the other paths. Each link is therefore only affected by the traffic from one of the dispersed sources, and this source can be seen as the fraction of traffic that the original source sends over that specific link. In order to obtain the same load on a link with as without dispersion, we assume that the link instead of carrying the traffic from a number of independent original sources now carries the traffic from N times as many independent dispersed sources. That is, one link carries fractions of the traffic from each of N times as many original and independent sources. This justifies the independence criterion used in the capacity calculations.

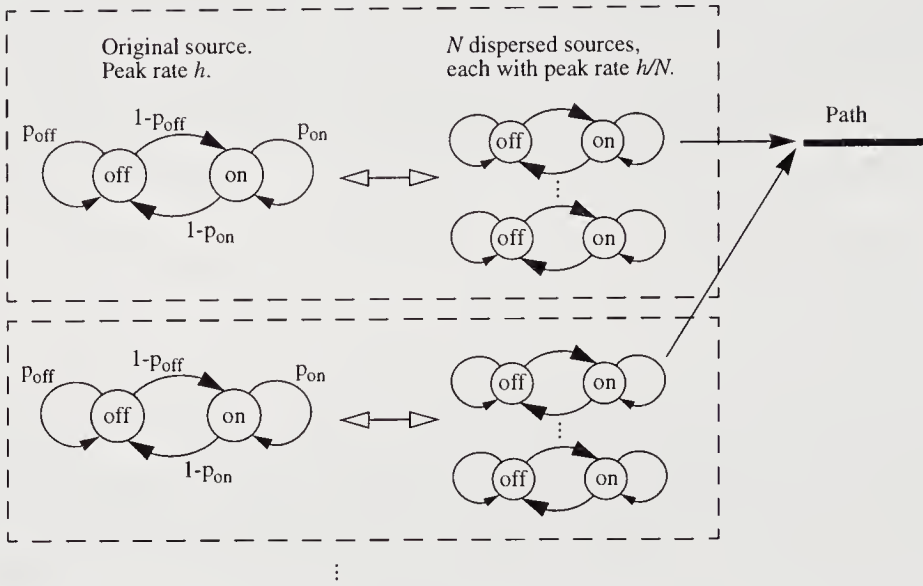


Figure 3 Modelling dispersed traffic sources. Without dispersion, a number of sources are multiplexed on a certain link, while in the case of dispersion, N times as many dispersed sources are multiplexed on the same link. The amount of traffic that the link carries is hence kept constant.

We can now calculate the equivalent capacity for a dispersed source, but we multiply it by N in order to get a fair comparison with the capacity for an original source. From (3), we get the cell-loss probability for n sources, each with peak rate h/N as

$$\varphi = \frac{1}{n\varepsilon} \sum_{x=\frac{NC}{h}}^n \left(x - \frac{NC}{h}\right) \binom{n}{x} \varepsilon^x (1-\varepsilon)^{n-x}. \quad (5)$$

For a given cell-loss probability, the capacity required for each source on each path is C/n , and the total capacity required for a source is NC/n . This is the capacity presented in the Figures.

Figure 4 shows for each value of N the aggregated equivalent capacity of N dispersed sources. The equivalent capacity for one source without dispersion is normalized to one. Note that this implies that the graphs for different values of C/h and burstiness are not directly comparable. The graphs only intend to show the multiplexing gain obtained by dispersion for different values of burstiness and source peak rate. The upper left graph shows a small increase in equivalent capacity for $N=2$ compared to $N=1$. The equivalent capacity for a dispersed source is in this case lower than the capacity for a non-dispersed source, but it still exceeds fifty percent of that value. When multiplied by two, it hence causes an increase in equivalent capacity. It should be noted that since the peak-to-link ratio is very high, only a few sources fit, given the zero-buffer assumption and the stringent loss requirement (10^{-9}). The peak in the graph is basically due to the fact that the number of sources has to be an integer. The truncation gives proportionally higher effect in this case since the number of multiplexed sources is very low (6 in the case without dispersion).

The figure shows that traffic dispersion decreases the equivalent capacity for a connection. When dealing with statistical multiplexing, the most troublesome traffic sources are those with a high peak-to-mean ratio and those with a high peak-to-link ratio. This is because it becomes extremely difficult to predict the amount of capacity which needs to be allocated in order to fulfil the performance requirements for heavily fluctuating sources. We can see that these sources are those where the benefits of traffic dispersion are most significant.

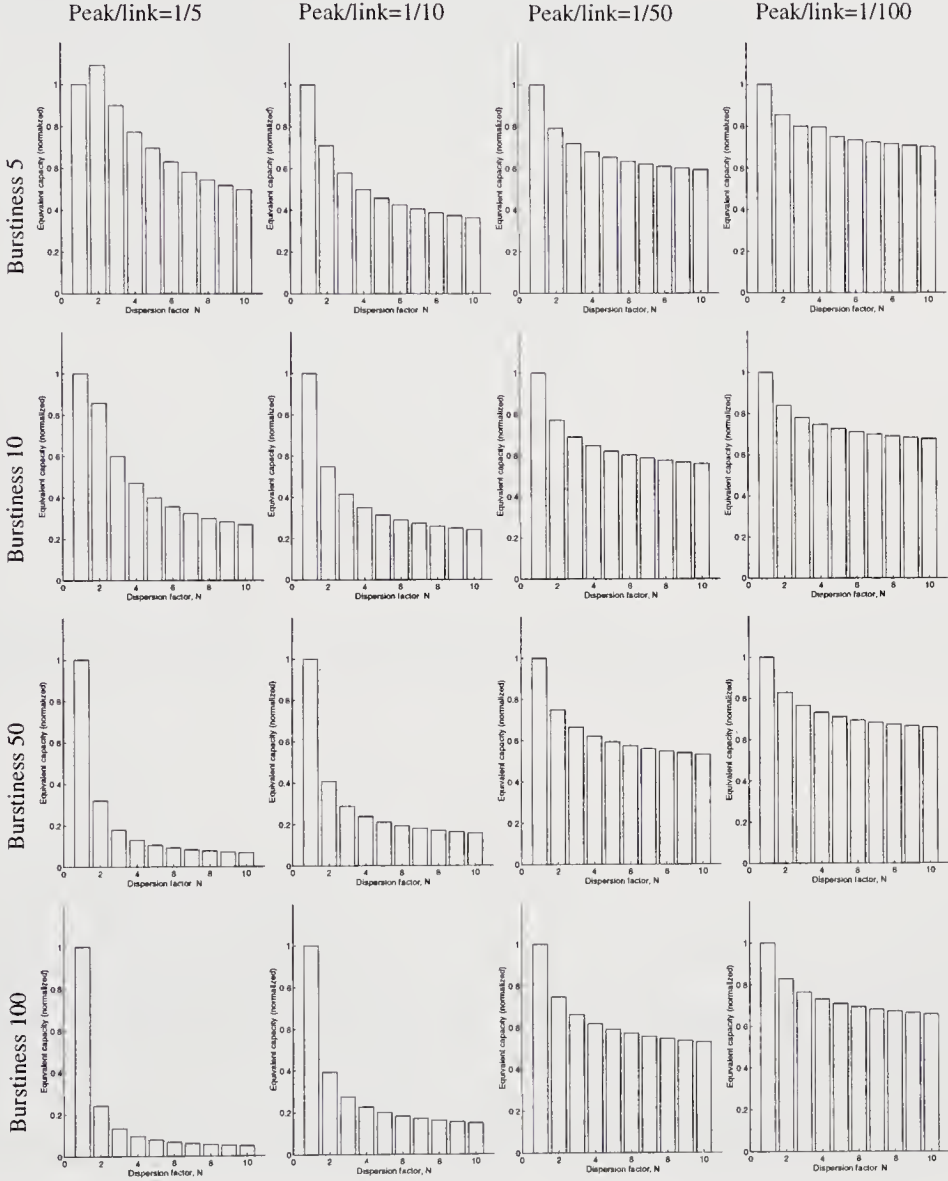


Figure 4 Equivalent capacity for different degrees of dispersion, and different values of source burstiness and peak-to-link ratio. The burstiness is defined as the source's peak rate divided by its mean rate, and 'Peak/link' denotes the source peak rate divided by the link capacity. The cell-loss probability was set to 10^{-9} .

In Figure 4, the values were normalized. Table 1 shows the equivalent capacity without dispersion ($N=1$) before normalization, given the link capacity $C=1$. This means that for a given column, the peak rate is constant, while the mean rate decreases for each row in order to make the source more bursty. For a given row, the peak rate as well as the mean rate decreases for each column, to make the source less dominant on the link. Table 2 shows the peak rates h and mean rates $\bar{\epsilon}h$ corresponding to the capacity values in Table 1.

A comparison of the two tables shows that the worst source behaviour is in the lower left box, while the best behaviour is in the upper right box. In the lower left box, the source burstiness is high, causing large fluctuations in the traffic, and the peak rate occupies a large part of the link capacity. The equivalent capacity for such a source is close to the peak rate. On the other hand, the source in the upper right box causes small fluctuations, never demanding more than a small fraction of the link, wherefore the equivalent capacity is close to the mean rate.

Table 1 Equivalent capacity before normalization; $C=1$, $N=1$, cell-loss probability 10^{-9}

$l/\bar{\epsilon}$	$h = 2.0 \cdot 10^{-1}$	$h = 1.0 \cdot 10^{-1}$	$h = 2.0 \cdot 10^{-2}$	$h = 1.0 \cdot 10^{-2}$
5	$1.7 \cdot 10^{-1}$	$9.1 \cdot 10^{-2}$	$8.3 \cdot 10^{-3}$	$3.3 \cdot 10^{-3}$
10	$1.7 \cdot 10^{-1}$	$7.1 \cdot 10^{-2}$	$4.5 \cdot 10^{-3}$	$1.7 \cdot 10^{-3}$
50	$1.4 \cdot 10^{-1}$	$2.3 \cdot 10^{-2}$	$9.6 \cdot 10^{-4}$	$3.6 \cdot 10^{-4}$
100	$1.0 \cdot 10^{-1}$	$1.2 \cdot 10^{-2}$	$4.8 \cdot 10^{-4}$	$1.8 \cdot 10^{-4}$

Table 2 Source mean rate

$l/\bar{\epsilon}$	$h = 2.0 \cdot 10^{-1}$	$h = 1.0 \cdot 10^{-1}$	$h = 2.0 \cdot 10^{-2}$	$h = 1.0 \cdot 10^{-2}$
5	$4.0 \cdot 10^{-2}$	$2.0 \cdot 10^{-2}$	$4.0 \cdot 10^{-3}$	$2.0 \cdot 10^{-3}$
10	$2.0 \cdot 10^{-2}$	$1.0 \cdot 10^{-2}$	$2.0 \cdot 10^{-3}$	$1.0 \cdot 10^{-3}$
50	$4.0 \cdot 10^{-3}$	$2.0 \cdot 10^{-3}$	$4.0 \cdot 10^{-4}$	$2.0 \cdot 10^{-4}$
100	$2.0 \cdot 10^{-3}$	$1.0 \cdot 10^{-3}$	$2.0 \cdot 10^{-4}$	$1.0 \cdot 10^{-4}$

It might also be interesting to study the influence of the cell-loss probability on the results. In the calculations discussed above, the cell-loss probability was kept constant and equal to 10^{-9} . Figure 5 shows that if we increase the cell-loss probability to 10^{-3} , traffic dispersion still reduces the capacity like in Figure 4, but not to the same extent as with the lower cell-loss probability. This might be because a higher tolerance of loss allows more sources to be multiplexed on the same link. The increase in number of sources, which dispersion makes possible, hence becomes less significant.

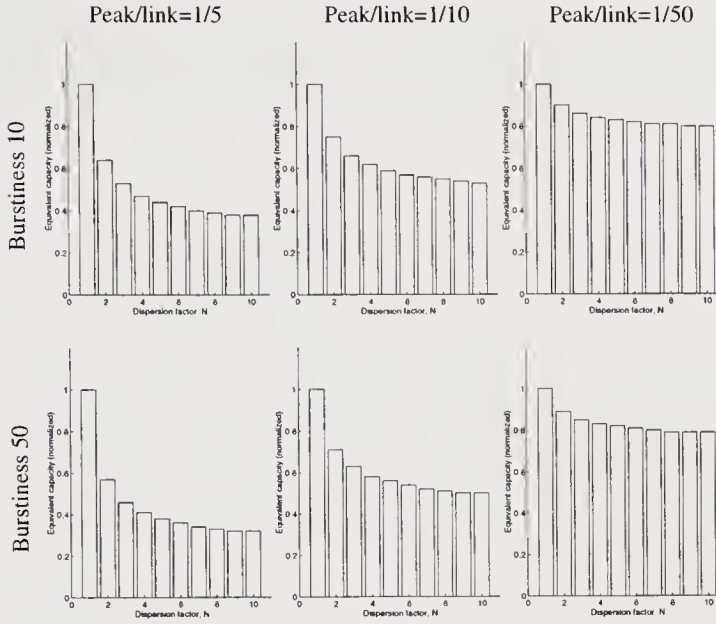


Figure 5 Equivalent capacity for some different values on source burstiness and peak-to-link ratio. The cell-loss probability was set to 10^{-3} .

2.2 Equivalent capacity with buffering

The discussion so far has been for a zero-buffer assumption, but it may also be of interest to look at the case where a single on-off source generates input traffic to a link with buffer capacity X . Guérin et al. give an upper bound on the equivalent capacity c of such a connection, Guérin et al. (1991):

$$c = h \cdot \frac{y - X + \sqrt{(y - X)^2 + 4X\bar{\epsilon}y}}{2y}, \text{ where } y = T_{on}(1 - \bar{\epsilon})h \cdot \ln \frac{1}{\phi}. \quad (6)$$

T_{on} is the average duration of an active period, ϕ is the probability of buffer overflow (cell loss), and h and $\bar{\epsilon}$ are defined as before. Since this case concerns only a single source, it might not be suitable to model dispersion as in the previous section, where each original source was replaced by a number of sources with lower peak rates. We therefore choose to model a dispersed source by an on-off source with a fixed mean but with a correlation function which changes with the dispersion factor.

Recall the on-off source from Figure 2. Define $u(i)$ to be the number of cells generated within the i th time unit. This means that $u(i)$ is either 0 or h . The correlation sequence of the source is given by

$$r(k) = E\{u(i+k)u(i)\} = h^2\bar{\epsilon}^2 \left(1 + \frac{1-p_{on}}{1-p_{off}} \cdot (p_{on} + p_{off} - 1)^k \right). \quad (7)$$

The more correlated the traffic is, the more difficult it becomes to handle. When dispersing the traffic, the objective is therefore to minimize the correlation in the cell stream on each path. As the correlation sequence of an on-off source is monotonously decreasing, the minimization is obtained by distributing the generated cells cyclically over the paths, as mentioned before (Figure 6).

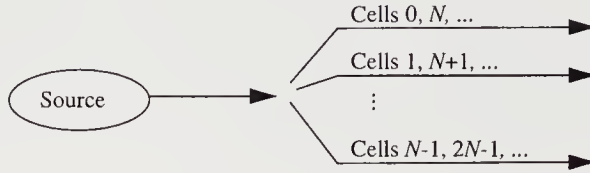


Figure 6 Dispersing the cells cyclically over N disjoint paths.

The correlation sequence between the cells on one of the paths is hence given by Gustafsson (1994:2)

$$r_d(k) = E \{ u(iN + kN) u(iN) \} = r(kN). \quad (8)$$

In order to study the behaviour of the equivalent capacity for different degrees of dispersion, we model the traffic from a dispersed source on a certain path. This is achieved by keeping the peak and mean rates of an on-off source constant, while reducing the correlation according to (8). The amount of traffic transmitted on a link during a certain time interval hereby remains unaltered, while the traffic behaviour varies under the influence of dispersion.

The fraction of time that the source spends in active state can be written as

$$\varepsilon = \frac{1 - p_{off}}{2 - p_{on} - p_{off}} = \frac{1}{1 + \frac{1 - p_{on}}{1 - p_{off}}}, \quad (9)$$

and keeping ε constant thus means keeping $\frac{1 - p_{on}}{1 - p_{off}}$ constant.

The only part of $r(k)$ varying when the source peak and mean rates are fixed, is hence $(p_{on} + p_{off} - 1)^k$.

Given the transition probabilities for a non-dispersed source ($N=1$), we can calculate the probabilities for dispersion factor N by

$$\frac{1 - p_{off}^{(1)}}{2 - p_{on}^{(1)} - p_{off}^{(1)}} = \frac{1 - p_{off}^{(N)}}{2 - p_{on}^{(N)} - p_{off}^{(N)}} \text{ and} \quad (10)$$

$$\left(p_{on}^{(1)} + p_{off}^{(1)} - 1 \right)^N = p_{on}^{(N)} + p_{off}^{(N)} - 1. \quad (11)$$

By adjusting the source characteristics according to (10) and (11), we show the effects of dispersion on the equivalent capacity from (6). We have chosen a numerical example with an on-off source whose T_{on} is about 200 time units.

Previous results have shown that the queue size is highly dependent on the correlation in the traffic, Li and Mark (1990). In order to facilitate a comparison among the different graphs, we therefore keep the correlation fixed in all cases where there is no dispersion, that is, the first bar in each graph. This means that the value of T_{on} is given by

$$T_{on} = \frac{1}{1 - p_{on}}, \quad (12)$$

will vary, because changing the burstiness while keeping the correlation fixed for $N=1$, means that the value of p_{on} will change as well.

We try to make the comparison among different traffic cases as fair as possible. One solution would be to keep the peak rate of the source constant. This means that the mean rate must be varied when the burstiness varies. The results obtained on these conditions are shown in Figure 7, and the equivalent capacity for $N=1$ is normalized to be one. The peak rate is constant and set to 100, and ϕ is 10^{-9} . Since the peak-to-link ratio is not considered in (6), the graphs in Figure 7 are not directly comparable to those in Figure 4. Additional results, which are not presented here, shows that the equivalent capacity behaves similarly if the mean rate is kept constant and the peak rate is changed instead, Gustafsson (1995).

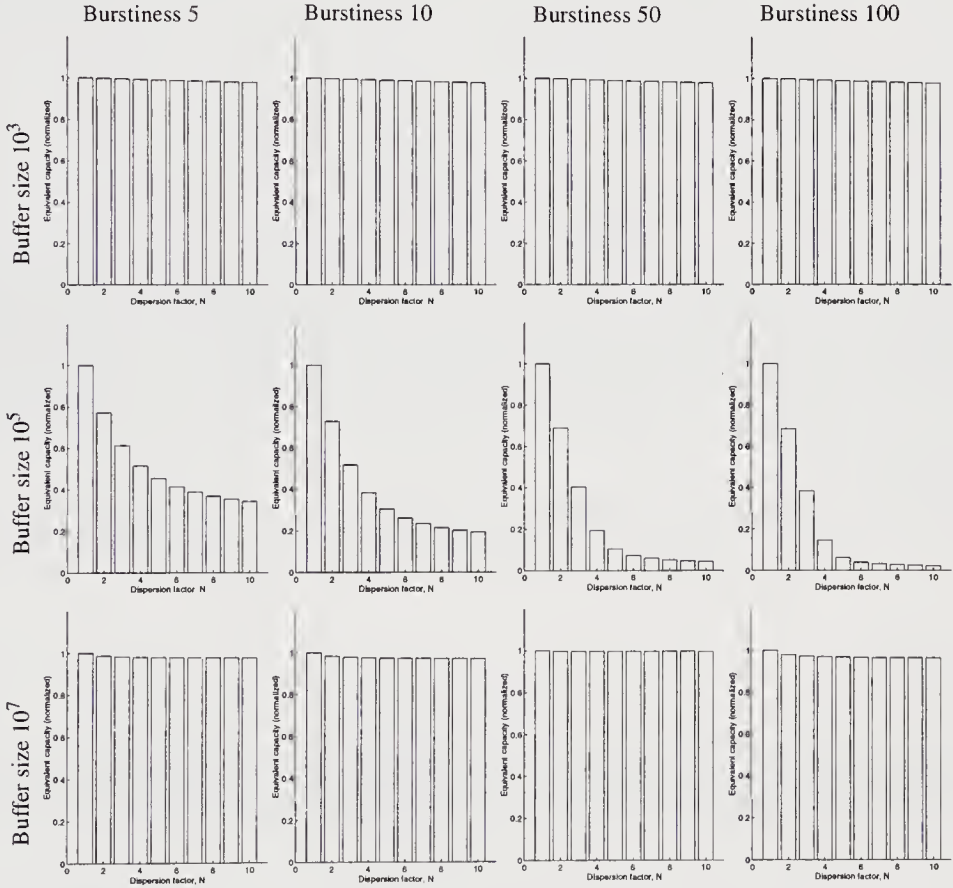


Figure 7 Equivalent capacity for different degrees of dispersion and burstiness, for three different buffer sizes. The source peak rate is constant and set to 100.

Table 3 shows the values from Figure 7 before normalization, for the cases without dispersion.

Table 3 Equivalent capacity before normalization; $N=1$, $h=100$

<i>Buffer size</i>	<i>Burstiness 5</i>	<i>Burstiness 10</i>	<i>Burstiness 50</i>	<i>Burstiness 100</i>
	<i>Mean rate 20</i>	<i>Mean rate 10</i>	<i>Mean rate 2</i>	<i>Mean rate 1</i>
	$T_{on}=248$	$T_{on}=220$	$T_{on}=202$	$T_{on}=200$
10^3	99.8	99.8	99.8	99.8
10^5	77.4	76.5	75.8	75.7
10^7	20.5	10.3	2.01	1.04

The results above show that when the buffer size is extremely small, the equivalent capacity for a connection approaches the source peak rate. Because of the buffer limitation, it becomes difficult not to exceed the allowed probability of overflow, even when dispersion is used. In the example above, the average burst size is about 20 000 cells, while the buffer size is only 1000. When a burst arrives, it hence fills up the buffer rather fast, and in order to keep the cell loss at a low level, the output rate of the buffer must be very high. By increasing the dispersion factor further, we can reduce the average burst size far below the buffer size, to a point where the traffic is almost completely uncorrelated, but it turns out that the decrease in equivalent capacity stays at about 45-50%. An explanation for this might be that the formula only considers a single source. There are hence no capacity gains due to the effects of multiplexing, as can be obtained when several sources share a link. This might also explain why the capacity reductions are not similar to the ones obtained without buffering, since in that case, multiple sources were multiplexed together on a link. In summary, with one single source and a small buffer, dispersion cannot significantly improve the situation, at least not for a dispersion factor smaller than ten.

If on the contrary the buffer size is extremely large, the equivalent capacity approaches the source mean rate. Since the capacity can never be lower than the mean rate, dispersion is of very little help in this case. It should be noted however, that such low capacity values can be obtained because the buffer is large enough to hold entire bursts. The penalty for this is long delays.

When the buffer size lies somewhere between these two extremes, traffic dispersion is useful, and the equivalent capacity under the influence of dispersion follows the same tendency with as without buffering. That is, as the source burstiness increases, the gain obtained by dispersion increases too.

Next, we consider the equivalent capacity of a number of connections, which are multiplexed on the same link. The capacity could be approximated by the sum of the individual capacities, that is

$$C = \sum_{i=1}^n c_i. \quad (13)$$

Unless the equivalent capacity of each individual connection is very close to the source mean rate, the capacity according to (13) in many cases overestimates what actually needs to be allocated. This is because the method does not consider the effects of statistical multiplexing.

Guérin et al. (1991) present the following approximation for the equivalent capacity of n multiplexed on-off sources:

$$C = \min \left\{ n \cdot \bar{\epsilon} h + \sigma \sqrt{-2 \ln \phi - \ln(2\pi)}, \sum_{i=1}^n c_i \right\}. \quad (14)$$

The term $n \cdot \bar{\epsilon} h$ denotes the mean aggregate bit rate of the connections, and σ is the standard deviation of the aggregate bit rate, that is

$$\sigma^2 = \sum_{i=1}^n \sigma_i^2 = n \sigma_i^2 = n \cdot h^2 \bar{\epsilon} (1 - \bar{\epsilon}). \quad (15)$$

As long as we keep the source peak and mean rates constant, the first part of (14) will not be affected by dispersion. In order to investigate the effects of dispersion on the equivalent capacity in (14), we therefore recall the model of a dispersed source which was used in Section 2.1. By replacing each original source with peak rate h by N sources, each with peak rate h/N , the first part of (14) becomes

$$N \cdot n \bar{\epsilon} \cdot \frac{h}{N} + \sigma \sqrt{-2 \ln \phi - \ln(2\pi)} = n \cdot \bar{\epsilon} h + \sigma \sqrt{-2 \ln \phi - \ln(2\pi)}, \text{ where} \quad (16)$$

$$\sigma^2 = N \cdot n \cdot \left(\frac{h}{N} \right)^2 \cdot \bar{\epsilon} (1 - \bar{\epsilon}) = \frac{n}{N} \cdot h^2 \bar{\epsilon} (1 - \bar{\epsilon}). \quad (17)$$

Figure 8 shows how dispersion affects the equivalent capacity of n multiplexed connections, according to (14). The results presented are for $n=10, 100, 1000$, and the values are normalized to be one for $N=1$. We have chosen the buffer size $X=100\,000$, since this was the case where dispersion made significant difference to the results in the previous discussion. Further results, which are not shown here, indicate that we get a capacity reduction as well when reducing the buffer size to about 1000, even though the reduction in that case is slightly smaller than with a larger buffer.

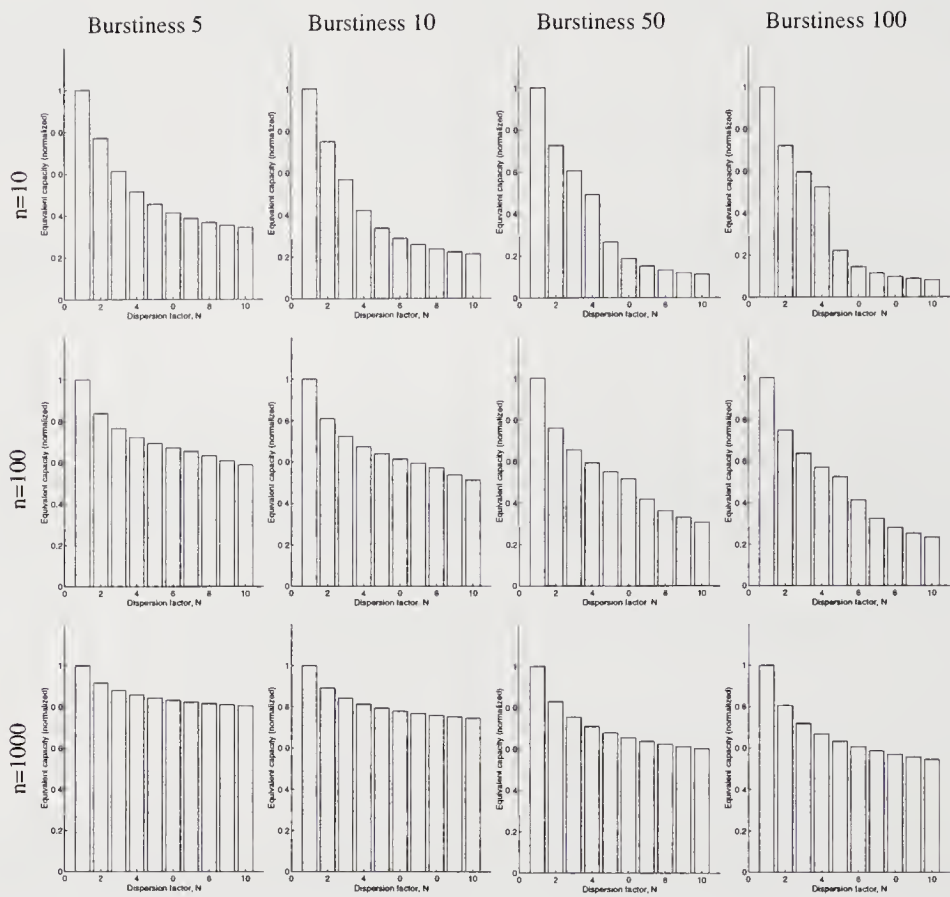


Figure 8 Equivalent capacity for different degrees of dispersion and burstiness. The peak rate of a source is constant and set to 100, the probability of buffer overflow ϕ is 10^{-9} , and the buffer size is $X=100\,000$.

Table 4 shows the equivalent capacity from Figure 8 before normalization for $N=1$. In the table, we show the aggregate equivalent capacity for n sources, divided by the number of sources, n . The values in the table can therefore be seen as the equivalent capacity of one of the n sources.

Table 4 Equivalent capacity before normalization; $N=1$, $h=100$

Number of sources, n	Burstiness 5	Burstiness 10	Burstiness 50	Burstiness 100
10	77.5	69.7	29.9	20.8
100	45.2	28.9	10.8	7.26
1000	28.0	16.0	4.79	2.98

Comparing these results to those in Figure 7 and Table 3, we find that (14) gives significantly lower capacity values than (13) for a large number of sources with high burstiness. This is the situation where the effects of statistical multiplexing show.

Furthermore, dispersion causes larger capacity reductions in the case with a high source burstiness, and a small number of sources which are multiplexed together. The similarity between this behaviour and the one that appeared in Figure 4 is striking. An increasing number of sources (larger n) means that the ratio between the source peak rate, which in this case is constant, and the aggregate equivalent capacity C decreases. If we let this ratio correspond to the peak-to-link ratio in Figure 4, we get exactly the same tendency with as without buffering. This means that we can make the general conclusion that dispersion improves the equivalent capacity particularly in the case of a small number of sources (high peak-to-link ratio) with high burstiness (high peak-to-mean ratio).

The results presented thus show that for a suitable buffer size, traffic dispersion reduces the equivalent capacity for a connection. For very large buffers dispersion does not affect the equivalent capacity, but will probably reduce the delay, and for very small buffers dispersion over a modest number of paths cannot improve the situation, unless there are enough sources to obtain multiplexing effects. When there is a capacity reduction, it behaves similarly with as without buffering. We have therefore chosen to limit the following discussions to the results without buffering, since they seem to represent a general behaviour.

3 COST FUNCTIONS

The previous section showed that a drastic decrease in equivalent capacity owing to traffic dispersion is possible for bursty sources. Considering only the equivalent capacity might however be somewhat optimistic, since spreading the traffic over several paths requires more virtual circuits to be established, and causes additional signalling overhead. We will therefore weigh the capacity obtained without buffering with three different cost functions, in order to establish under what circumstances traffic dispersion is profitable.

The first cost function is a fixed charge per capacity unit (Figure 9 (a)). This cost is independent of the number of connections used for a transmission, and the cost benefit curve will follow the curves in Figure 4, scaled by a constant cost factor.

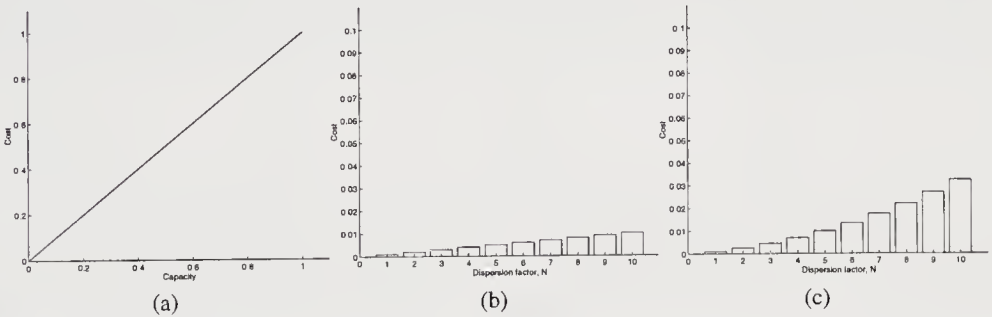


Figure 9 The different costs considered: a fixed cost per capacity unit (a), a fixed cost per connection (b), and a cost increasing with the number of connections (c).

Next, we consider a cost function which is composed of a fixed charge per capacity unit, and a fixed charge per connection used for a transmission (Figure 9 (a) and (b)). The connection charge is motivated by the extra effort needed to set up and maintain several virtual circuits for each transmission.

The last cost function is composed of a fixed charge per capacity unit, and a progressively increasing charge per number of connections used (Figure 9 (a) and (c)). Assume that without dispersion, the virtual circuit follows the shortest path through the network. Since there might only be one path of that length, the additional connections needed for dispersion will have to follow longer paths. The cost increase could therefore be taken as a penalty for using longer and longer paths.

4 WHEN IS TRAFFIC DISPERSION USEFUL?

We relate the different cost functions to the values of equivalent capacity that we obtained in Section 2.1, to see whether dispersion is always motivated. If we only consider a fixed charge per capacity unit and assume that there is no extra cost for using several connections (the first cost function), traffic dispersion is practically always profitable, and the more paths used the better. For sources with a high burstiness and a high peak-to-link ratio, the benefits of dispersion are obvious; by spreading the traffic over only a handful of paths, a cost benefit of about eighty to ninety percent is obtained. Regarding the sources with a low peak-to-link ratio, the benefits are not that large. The gain here is only about thirty percent. However, when considering the values of equivalent capacity without normalization, in Table 1, we find that the cases where the benefits of dispersion are least significant, are those where the cost without dispersion is already very low. Any larger gain would therefore in real values be negligible in comparison to the other cases. In other words: when the gain is needed, it is high.

With this cost situation, traffic dispersion over many paths is consequently always the best solution. The assumption of no extra cost for extra paths may however not be quite realistic, wherefore we move on to the next cost function.

In this case, we have a fixed charge per capacity unit and a fixed charge per connection. The balance between these two charges is very important. If the charge per link is extremely small, the results are the same as above, which means that dispersion is always profitable. If, on the other hand, the charge per link is too large, it will dominate the total cost and result in a cost function which is linearly increasing with the number of paths. Traffic dispersion would hence never be justified.

More realistically, the charge per capacity unit will form the major part of the cost. In our calculations, we have chosen the cost 1 per capacity unit and 0.001 per connection, as a hopefully reasonable proportion. These values apply to a time unit equal to one call. Figure 10 shows the result from applying such a cost function on some of the equivalent capacity values from Figure 4, without normalization.

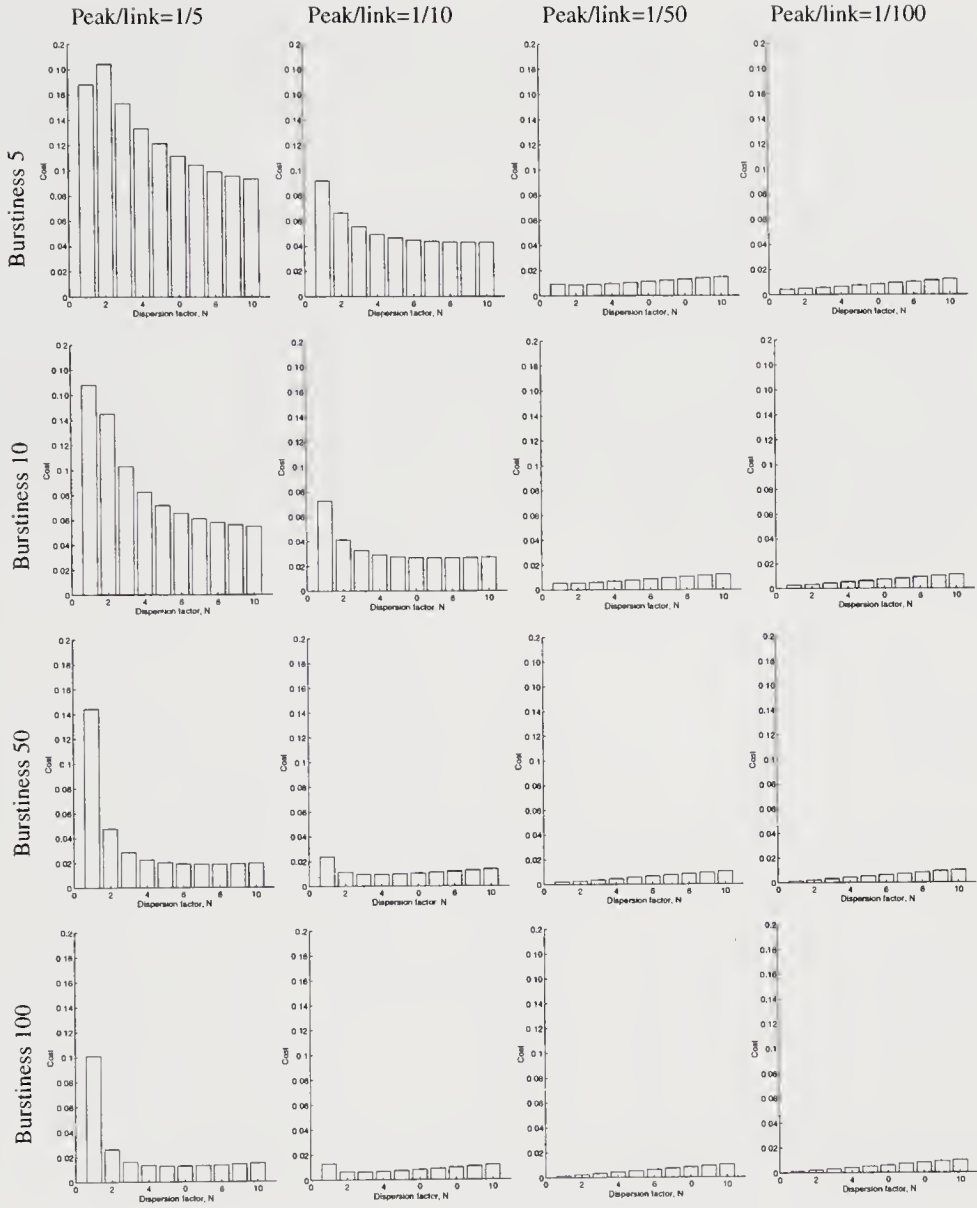


Figure 10 The second cost function related to the equivalent capacity values from Figure 4. The cost is 1 per capacity unit and 0.001 per connection.

These results show again that traffic dispersion is very profitable in the cases where the peak-to-link ratio is high. The maximum gain is about eighty percent for sources with high burstiness. As the peak-to-link ratio decreases, so does the gain, and using a larger number of connections even causes a small cost increase.

The conclusion is that in cases which can be handled well without dispersion, it should not be used. It is then better to allocate resources in a more conventional manner, using only one path for each transmission. In the cases where traffic dispersion does give benefits, it should of course be used. The results show that spreading the traffic over more than about two to five paths does not give any remarkable further benefit, whereas the number of paths should preferably be kept to about this size.

With the chosen proportions on the cost function, the increased cost caused by several connections is however practically negligible compared to the gain in cost obtained on other conditions. In hesitation of whether dispersion should be used or not, it therefore seems better to use it, since the penalty for dispersing when unnecessary is very small compared to the gain obtained when dispersion turns out to be needed. In essence, the benefits from using dispersion in the right place are many times larger than the penalty for using it in the wrong place.

The last cost function is a fixed charge per capacity unit and a charge which increases with the number of connections. The behaviour is as the one we described previously, namely the charge for using several paths soon dominates the total cost, and a transmission will quickly become rather expensive. This is shown in Figure 11. In this example, considerable benefits are still obtained for the higher peak-to-link ratios, but in the other cases there is no gain at all. The best strategy under these circumstances seems to be to disperse the traffic sparingly, and only when an economic gain is guaranteed.

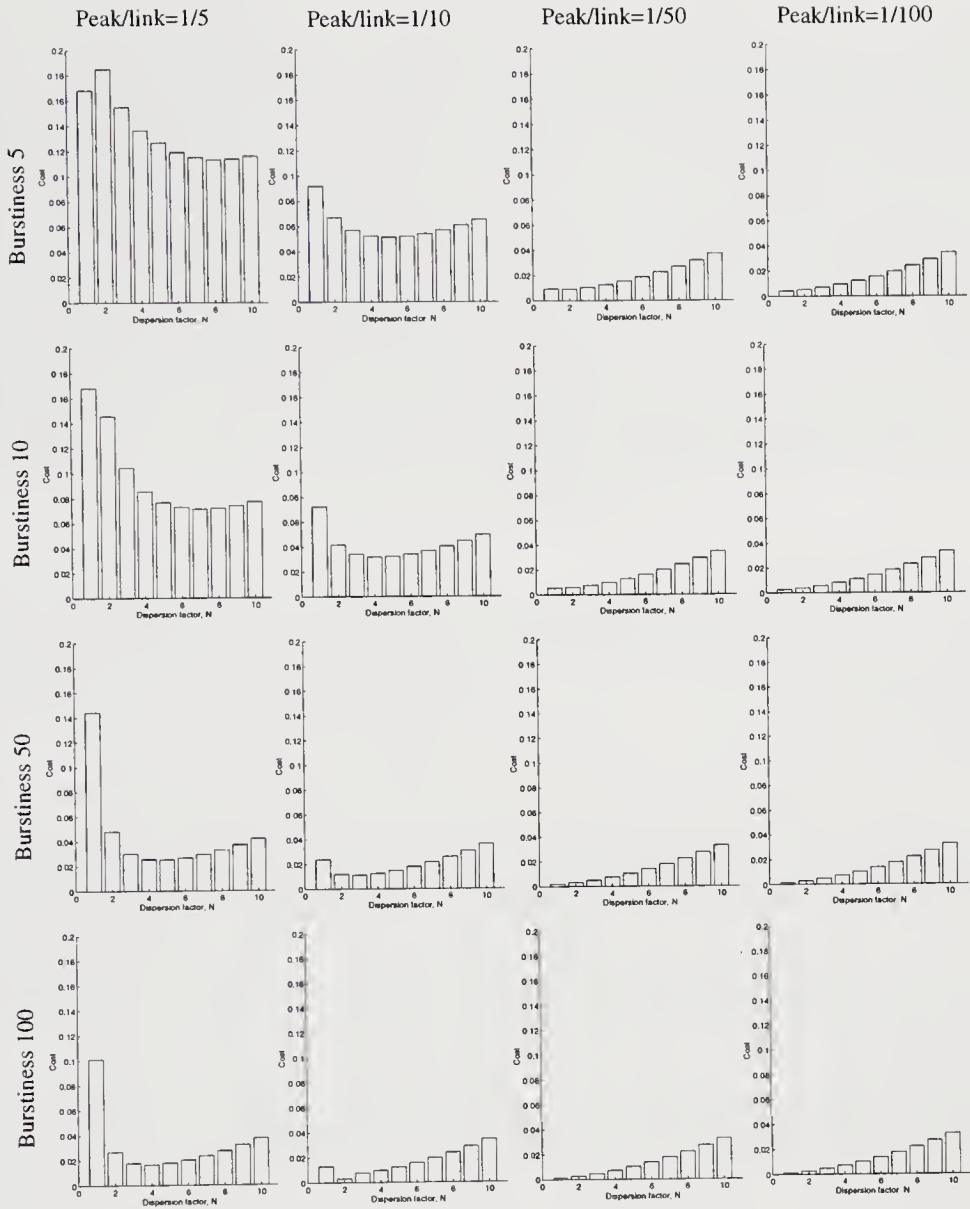


Figure 11 The third cost function related to the equivalent capacity values from Figure 4.

So, when is traffic dispersion useful? On the condition that the penalty for using several connections does not dominate the total cost for a transmission, dispersion over a moderate number of paths is practically always profitable. The most troublesome traffic sources to be handled by statistical multiplexing - that is, as mentioned before, those with a high burstiness and a high peak-to-link ratio - are those for which the highest gains can be made.

ATM Forum has defined three traffic parameters, namely sustained bit rate, peak bit rate and burst size. As a rough estimation, dispersion should be employed when the relation between the peak bit rate and the sustained bit rate (the source burstiness) is in the order of ten or more, and when the peak bit rate exceeds one tenth of the link capacity. In all cases, the number of paths used for a transmission should stay somewhere between two and five. The burst size is not directly considered in this paper. It is true that the average duration of an active period is obtained directly from the transition probabilities of an on-off source, and so is the probability that the source is in active state. We have however only considered the probability of being in active state, and it is possible to change that probability without affecting the duration of the active period. However, longer bursts (longer active periods) indicate higher correlation in the traffic, and this is where the benefits from dispersion are indisputable, Gustafsson (1994:2).

When dealing with sources having low peak bit rate compared to the sustained bit rate, that is, less than a ratio ten to one, and a link capacity above ten times the peak bit rate, we might just as well do without dispersion. The penalty for using dispersion in vain is however not dramatic, and dispersion may be applied in uncertain cases. Lastly, it should be noted that users may want to pay extra to get the traffic dispersed for reasons of security, or other reasons that are not contained in the results presented in this paper.

5 CONCLUSIONS

This paper presents spatial traffic dispersion as a means for handling difficult traffic sources, and facilitating resource allocation. The use of dispersion shows a large gain in the equivalent capacity, and when relating the capacity to three different cost functions, the benefits are in most cases confirmed.

From the results presented, we conclude that a profit due to dispersion is practically always possible. In the case of a single traffic source, there are no multiplexing effects. On the one hand, a small buffer may limit the gain in equivalent capacity to an extent where dispersion over a modest number of paths cannot improve the values. On the other hand, a large buffer makes the equivalent capacity close to the mean without dispersion, at the expense of long delays. In this case, dispersion could probably reduce the delay, but it is not reflected in the capacity results.

Furthermore, we conclude that the cost benefits from using dispersion are most important for sources with a high peak-to-mean ratio (larger than ten), and a high peak-to-link ratio (larger than one tenth). This is on condition that the charge per capacity unit dominates over the cost for using several connections. The penalty for using dispersion when not necessary turns out to be small compared to the benefits from using dispersion where it is really needed. Traffic dispersion is therefore useful in all cases where its benefits are beyond all doubt, as well as in all uncertain cases.

We may also change our viewpoint from the user to the network operator. If a tariff structure is imposed that erroneously penalizes traffic dispersion, statistical multiplexing may not be used to its full potential in the network. For a user it namely means a charge according to behaviour and not to average use, since the equivalent capacity of the transmission is strongly dependent on the burstiness of the source. The consequence is that the user may choose to keep the connection for a shorter time, and set it up for individual bursts. The operator thus has a situation with low sharing of resources (fewer paying users simultaneously connected) and with substantially more connection requests. As our example of equivalent capacity shows, traffic dispersion basically makes the statistical link sharing immune to source behaviour. We therefore hope that it will be widely employed as an antidote to new bursty traffic sources.

6 REFERENCES

- Cheng, T-H. (1994) Bandwidth allocation in B-ISDN. *Computer Networks and ISDN Systems*, Vol. 26, No. 9, 1129-42.
- Déjean, J.H., Dittmann, L. and Lorenzen, C.N. (1991) String Mode - A New Concept for Performance Improvement of ATM Networks. *IEEE Journal on Selected Areas in Communications*, Vol. 9, No. 9, 1452-60.
- Guérin, R., Ahmadi, H. and Naghshineh, M. (1991) Equivalent Capacity and Its Application to Bandwidth Allocation in High-Speed Networks. *IEEE Journal on Selected Areas in Communications*, Vol. 9, No. 7, 968-81.
- Gustafsson, E. (1994) *Traffic Dispersion - A Literature Survey*. Technical Report TRITA-IT R 94:35, Royal Institute of Technology, Stockholm.
- Gustafsson, E. (1994) *Traffic Dispersion in ATM Networks*. Technical Report TRITA-IT R 94:36, Royal Institute of Technology, Stockholm.
- Gustafsson, E. (1995) *When Is Traffic Dispersion Useful? A Study On Equivalent Capacity*. Technical Report TRITA-IT R 95:17, Royal Institute of Technology, Stockholm.
- Lee, T.T. and Liew, S.C. (1993) Parallel Communications for ATM Network Control and Management. *Proc. IEEE GLOBECOM*, Vol. 1, 442-6.
- Li, S-Q. (1989) Study of Information Loss in Packet Voice Systems. *IEEE Trans. on Communications*, Vol. 37, No. 11, 1192-1202.
- Li, S-Q. and Mark, J.W. (1990) Traffic Characterization for Integrated Services Networks. *IEEE Trans. on Communications*, Vol. 38, No. 8, 1231-43.
- Maxemchuk, N.F. (1975) Dispersy Routing. *Proc. of ICC '75*, San Fransisco, CA, 41-10-13.
- Maxemchuk, N.F. (1993) Dispersy Routing in High-Speed Networks. *Computer Networks and ISDN Systems*, Vol. 25, No. 6, 645-61.

7 BIOGRAPHIES

Eva Gustafsson received the M.Sc. degree in electrical engineering from the Royal Institute of Technology, KTH, in 1992. She is currently a Ph.D. student at the Department of Teleinformatics, working on traffic dispersion in high-speed networks.

Gunnar Karlsson received the MS degree from Chalmers University of Technology in 1983, and the Ph.D. from Columbia University in 1989. He was a Fulbright scholar at the University of Massachusetts at Amherst in 1982-83. His Ph.D. thesis was on sub-band coding of video for ATM networks. He joined the IBM Zurich Research Laboratory in 1989 and the Swedish Institute of Computer Science in 1992. He has been the first project leader of the Stockholm Gigabit Network. His research interests include traffic control, switching architectures and packet video. He is Docent at the Department of Teleinformatics at the Royal Institute of Technology (KTH).

PART THREE

Routing and Optimisation

A comparison of pre-planned routing techniques for virtual path restoration

P.A. Veitch and D.G. Smith

*Dept. of Electronic & Electrical Engineering,
University of Strathclyde,
Glasgow G1 1XW,
Scotland, U.K.*

Tel: (0141) 552 4400

Fax: (0141) 552 4968

E-mail: {pveitch,g.smith}@comms.eee.strathclyde.ac.uk

I. Hawker

*BT Laboratories,
Martlesham Heath,
Ipswich IP5 7RE,
England, U.K.*

Abstract

Network restoration techniques will be vital to ensure B-ISDN service survivability in the event of high capacity link and node failures. Reliable ATM crossconnect networks can be implemented by the strategic pre-assignment of protection Virtual Path (VP) routes to permit recovery from a realistic subset of all possible failures, eg single span failures. The method of protection route assignment influences the quantity of redundant resources like spare capacity and Virtual Path Identifiers (VPIs), whilst nodal hardware costs are incurred due to the requirement of pre-stored alternate routing information. In addition to implementation costs, the impact that the choice of rerouting scheme has on other factors must be considered. For example, the degree of path elongation following restoration may adversely affect the delay performance of certain connections. Also, the amount of computation required to design the protection routes, and the effort needed to activate such routes have to be taken into account. This paper formulates metrics to facilitate a comparative evaluation of four distinct routing strategies for VP restoration, and in conjunction with a discussion of qualitative properties of each scheme, it concludes that failure independent rerouting is the preferred approach.

Keywords

ATM Virtual Paths, restoration, routing, survivable network design

1 INTRODUCTION

Because the potential repercussions of a cable break or node failure in a high capacity broadband trunk network are so great, survivability is crucial (Wu, 1992). Restoration is the process of re-establishing trunk groups affected by a failure by exploiting spare capacity at diverse locations in a mesh topology (Veitch et al, 1995b). This is realised by high speed Digital Crossconnect Systems (DCSs) which are managed centrally, but also have the capability to interact in a distributed fashion, enabling fast restoration. If restoration is rapid enough, active calls may not be dropped. Indeed, a target completion time of 2 seconds would ensure preservation of the majority of voice connections (Sosnosky, 1994). Recent research into ATM Virtual Path (VP) restoration suggests that progress can be made in achieving very fast service recovery (Kawamura et al, 1994, Anderson et al, 1994, Veitch et al, 1995c). This is largely attributed to the logical nature of a Virtual Path which decouples routing and capacity assignment making reconfiguration simple compared with Synchronous Transfer Mode (STM) paths (Sato et al, 1990). This paper focuses on VP restoration in ATM networks, and in particular, the range of approaches to pre-planning alternate routes for this purpose.

From a network operator's point of view, a restoration strategy should be simple to implement, and resource efficient. In tandem with these requirements, the scheme should offer the subscriber fast service recovery from a wide range of failures. A suitable approach therefore, is to pre-assign restoration paths in advance of failure occurrence. This can be performed by a centralised computer with a global view of the network; resources can be managed efficiently and an appropriate subset of failures can be selected as the basis for protection. In the event of a failure, distributed signalling between crossconnects can be used to achieve very fast restoration with a simple protocol since it is only necessary to activate pre-determined routes. Two distinct methods of establishing protection VP routes which have been identified in the literature are categorised as failure dependent (Anderson et al, 1994) and failure independent (Kawamura et al, 1994) rerouting, both of which are defined later. Although both of these techniques constitute pre-assigned VP restoration, they are fundamentally different in certain aspects of implementation and performance, hence it is vital to perform a formal comparison. We focus on single span failure which is the simultaneous failure of all the transmission systems between two crossconnect nodes. This assumption facilitates a fair comparison of schemes, since the description of one of the two paradigms studied accounts for span failures only (Anderson et al, 1994).

The costs of implementing a particular restoration system are affected by the required spare resources such as link/buffer capacity and Virtual Path Identifiers (VPIs), as well as the memory overheads required to support the pre-storage of alternate routing information. In addition, different rerouting techniques can be assessed in terms of the computational effort required to design the protection plans, as well as the signalling effort needed to activate protection routes. With respect to performance, the choice of rerouting strategy affects the user-perceived quality of service, since restoration often induces path elongation, causing increased delays and cell delay variation. Following a

simple description of the alternative rerouting schemes in section 2, a comparative evaluation will be carried out in section 3 using metrics based on required spare capacity, VPI redundancy, path length elongation, storage overheads and the computational effort ascribed to protection route design. Section 4 discusses other qualitative factors that can be employed to compare the two distinct rerouting paradigms, including the signalling protocol and robustness in the presence of uncertainty. Section 5 concludes the paper by reasoning in favour of one particular method by taking into account all the quantified metrics of section 3 as well as the implementation aspects considered in section 4.

2 ALTERNATE VP REROUTING SCHEMES

Prior to providing a comparative evaluation, three failure dependent rerouting policies will be described, followed by an overview of the failure independent rerouting algorithm. Throughout, it is assumed that the working VP configuration is known a priori and that single span failure protection is required.

2.1 Failure dependent approaches

With failure dependent rerouting, each possible single span failure is examined in turn, and alternative routes are subsequently found for all the VPs affected by the failure. A batch alternate route planning operation of this kind may be written:

```

For each possible span failure
  For each failed VP
    Find alternate path according to rerouting policy
  End For
End For

```

Hence, there is a unique reconfiguration associated with each failure. Alternate routing data in the form of VPI and link ID information are stored in databases of all the relevant crossconnect nodes. When a span fails, the ID of the failed span is broadcast to all network nodes which re-load their lookup tables with the relevant data, resulting in an asynchronous logical (i.e. VP) topology update (Anderson et al, 1994). Three separate versions of failure dependent rerouting will now be explained.

2.1.1 Local rerouting

In this scheme, when a span fails, all the affected VPs are rerouted between the terminating nodes of the span, without regard to the source and destination of the VPs (Figure 1 (a)).

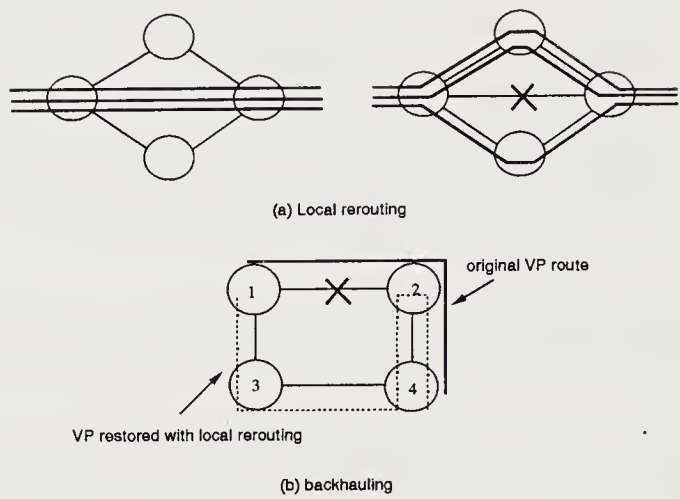


Figure 1: Characteristics of local rerouting

Although extremely simple to compute the alternate routes, and execution of restoration is potentially fast due to the majority of message processing being carried out in the vicinity of the failure, this is a greedy algorithm and can cause the undesirable phenomenon known as *backhauling* (Figure 1 (b)). From the figure, failure of span 1-2 leads to an alternate route being computed between nodes 1 and 2 as 1-3-4-2. The failed path 1-2-4 consequently uses the route 1-3-4-2-4 meaning span 2-4 is utilised twice.

2.1.2 Local-destination rerouting

A potentially more efficient result should be possible with a more sophisticated algorithm, such as “local-destination” rerouting proposed by AT&T (Anderson et al, 1994). Considering a span failure, failed VPs will be rerouted with one of the span terminating nodes as the starting point. The destination of the alternate route will depend on the individual VP route however, so as to reduce resource consumption. In Figure 2, if span 3-4 fails, then devising a shortest hop path between node 3 and the VP terminating node 8 will produce the two hop detour 3-7-8; a more efficient result than *pure* local rerouting. The essence of the algorithm is to retain as large a portion of the original working path route as possible, then find the most direct path to the destination of the failed VP whilst avoiding the failed span. From this very basic analysis therefore, a set of heuristics can be devised with respect to an individual VP affected by failure of a span:

1. The starting point of the detour is the terminating node of the failed span at the side of the VP with most hops: if equal, select at random.
2. Find the shortest hop path between starting point of detour and VP destination node.

3. Add retained part of path.
4. Remove self-loops and discount overlapping resource demands.

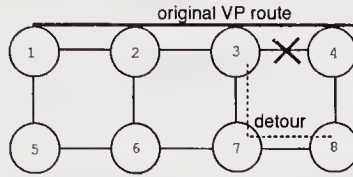


Figure 2: Characteristics of local-destination rerouting

Step 4 is included because backhauling is still possible, as can be seen in Figure 3 by the failure of span 2-3. The path is of equal length on each side of the failed span (1 hop). The starting point of the detour is selected as node 3, so we retain hop 6-3 of the original VP route, and seek a shortest hop path between node 3 and the VP termination, which is node 1. The result of the detour is thus 3-6-5-2-1, and when we concatenate this with the retained part of the original path, we obtain 6-3-6-5-2-1. Obviously, backhauling has occurred due to the self-loop 6-3-6, so this is eliminated leaving 6-5-2-1 as the new VP route employed due to failure of span 2-3. A final check to be made is whether or not the alternate route uses any of the original VP route hops; this occurs in the example with respect to span 1-2, hence the spare capacity/VPI requirements for this hop are discounted.

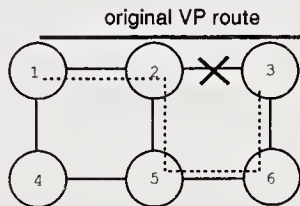


Figure 3: Self-loop and overlap during local-destination rerouting

2.1.3 Source-based rerouting

Source-based rerouting ought to be yet more efficient in terms of spare capacity (Anderson et al, 1994), by allowing alternate routes to be computed between the terminating nodes for each path affected by a failed span (Figure 4). The effect of this algorithm is to spread the demand for spare capacity more freely throughout the network than the preceding two schemes described. Any overlap between the original path and designated alternate route (eg span 1-2 of Figure 4) is dealt with by discounting the spare resource demands for such spans.

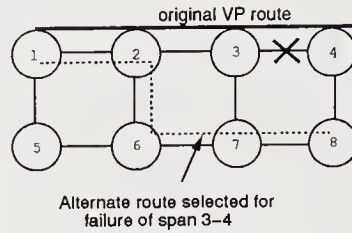


Figure 4: Source-based rerouting

2.2 Failure independent approach

In the failure independent case, a single alternate VP route can be designed to protect a working VP from any single span failure. The design criterion to satisfy this requirement is that a span disjoint route be selected for protection. Regardless of the underpinning physical span which induces VP failure therefore, the same protection route is employed for restoration. From Figure 5, whether span 1-2, 2-3 or 3-4 fails, the same backup route, 1-5-6-7-8-4 protects the working path 1-2-3-4. Indeed, failure of nodes 2 or 3 may be circumvented by activating this same route. Because there is a single alternate protection route for a working VP, the backup can be established in advance of failure by setting VPIs at the appropriate crossconnect nodes; from Figure 5, this corresponds to nodes 5, 6, 7 and 8. Activation of such a VP may be performed by altering the routing table at the connection endpoints (i.e. nodes 1 and 4 from the Figure). It is at such nodes that storage of alternate routing data is required.

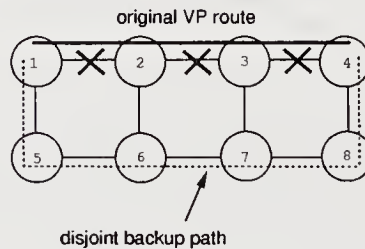


Figure 5: Failure independent (span disjoint) rerouting

The algorithm may be written:

```

For each VP
  Find shortest hop span disjoint path
End For
  
```

At the heart of this, and all of the failure dependent strategies described previously, is a shortest path computation. The common algorithm employed is the Floyd-Warshall technique, based on distance matrix manipulation. A very simple modification is made in that given the choice of two equidistant path routes, a random selection between the two

is made. It should be reiterated that for all the rerouting schemes investigated, shortest hop paths are found based on link weights of unity. On a batch provisioning basis, this produces suboptimal results in terms of spare capacity. If minimisation of the global spare capacity is required, a more complex solution to the rerouting problem would be required, such as mathematical programming or stochastic techniques based on simulated annealing (Coan et al, 1991). The techniques employed for the comparative evaluation detailed in the following section are not optimised in terms of spare capacity requirements so as to ensure that none of the other metrics which are quantified become negatively biased.

3 COMPARATIVE EVALUATION

3.1 Network assumptions

A variety of network models will be used to generate performance data for each of the four VP rerouting methods. Some pre-requisites are essential to simplify the analysis. The networks are meshed backbones comprising VP crossconnects, each of which is assumed to be collocated with a VC switch. Measurements of required resources (spare capacity and VPIs) and path lengths correspond to the inter crossconnect spans, not the links between VC and VP switching elements. Each span between crossconnect nodes will carry just one bidirectional fibre transmission system, enabling simple computation of redundant resources. In each network considered, a single bidirectional VP of unit capacity is established between each node pair, hence in an n node network, there are $n(n-1)/2$ Virtual Paths. These working paths will be generated using the Floyd-Warshall algorithm with shortest hop routes being selected. Alternate routing information which forms the basis of the design metrics for comparison, is then produced for each of the four schemes detailed in the previous section. Full protection from single span failures is provided in each case.

Prior to the analytic detail of individual metrics, some basic nomenclature is introduced. The physical network is described as a graph $G(V, E)$, whereby V is the set of vertices representative of ATM nodes, and E is the set of edges representing inter-nodal spans. A single vertex is denoted v ($v \in V$) whilst a single edge is symbolised as e ($e \in E$). The working capacity of an edge e is denoted W_e , whilst the spare capacity is S_e . The logical network is described by the set of paths P , whereby a single path π ($\pi \in P$) is the collection of edges traversed. The capacity of a path π is C_π . The set of protection routes is defined as \hat{P} , with a protection path pertaining to a working path π denoted $\hat{\pi}$ in the failure independent case and $\hat{\pi}^f$ in the failure dependent case, with the superscript f denoting the index of the failed edge, e_f . Note that $\hat{\pi}^f$ represents the end-to-end route of the path π following restoration from failure of edge e_f , part of which is often unchanged. For clarity, we further define $\hat{\pi}_d^f$ to be the edges of the rerouted part of the path only (the subscript d refers to *detour*). There are m edges, n vertices and k paths in the network. Additional notation will be introduced where necessary.

3.2 Computation of metrics

Given the working VP and alternate routing information, the following metrics can be computed for each of the four rerouting schemes applied to several network topologies.

3.2.1 Spare Capacity Ratio (*SCR*)

Ultimately, the *SCR* is the ratio of the aggregate spare capacity in the network to the aggregate working capacity. The working capacity of an edge is found by summing the capacities of constituent paths:

$$W_e = \sum_{\pi \in P, e \in \pi} C_\pi. \quad (1)$$

Hence, the total working capacity is found by summing (1) over the set of network edges. Computing the individual spare capacity quotas per edge is a little more complex. Depending on the edge which has failed in the network, the demanded spare capacity on the remaining edges differs. This is because a different set of working paths will be affected by each possible failure, hence a different reconfiguration is performed in each case. Letting S_e^f symbolise the spare capacity required on edge e due to failure of edge e_f , we have:

$$S_e^f = \sum_{\pi \in P, e_f \in \pi, e \in \pi} C_\pi. \quad (2)$$

For the failure independent case, or:

$$S_e^f = \sum_{\pi \in P, e_f \in \pi, e \in \hat{\pi}_d^f} C_\pi. \quad (3)$$

Which applies to the failure dependent case. Now, the provisioning of spare capacity on each edge must account for the edge failure which will yield the greatest demand for rerouted traffic. We thus find the required spare capacity for an edge e , denoted S_e , in the following way:

$$S_e = \max \{S_e^1, S_e^2, \dots, S_e^m\}. \quad (4)$$

It should be stressed that no attempt is made at capacity modularisation so as to conform to specific transmission systems. The value of *SCR* is subsequently found by dividing the total spare capacity by the total working capacity:

$$SCR = \frac{\sum_{e \in E} S_e}{\sum_{e \in E} W_e}. \quad (5)$$

3.2.2 Mean VPI Redundancy (*MVR*)

When protection routes are designed, Virtual Path Identifiers (VPIs) must be reserved for the appropriate links. The total number of idle VPIs is a function of the number of edges used in each protection route since VPI translation is performed for each link of a VP connection (ITU-T, 1993a). For the failure independent case, letting $L(\hat{\pi})$ be the length (number of edges used) of a specific protection path, $\hat{\pi}$, the total number of idle VPIs, denoted N^\vee , is found from:

$$N_{fi}^\vee = \sum_{\hat{\pi} \in \hat{P}} L(\hat{\pi}). \quad (6)$$

The subscript *fi* indicates *failure independent*. In a similar fashion, *fd* will specify the *failure dependent* version of appropriate metrics. For the failure dependent case, $L(\hat{\pi}_d^f)$ is the number of spans in the rerouted part of the end-to-end working path π , activated due to failure of e_f . Considering all failures per path, and the complete set of paths in the network:

$$N_{fd}^\vee = \sum_{\pi \in P} \sum_{e_f \in \pi} L(\hat{\pi}_d^f). \quad (7)$$

Now, given the total number of VPIs, regardless of the rerouting scheme, the *MVR* may be found by dividing the appropriate N^\vee by m (the number of edges) giving the mean VPI redundancy per edge; this quantity can then be normalised to the maximum number of VPIs per link (4096), yielding:

$$MVR = \frac{N^\vee/m}{4096}. \quad (8)$$

3.2.3 Path Elongation Factor (*PEF*)

This is simply the ratio of the mean length of a VP rerouted during failure restoration to the mean working path length. For the failure independent scheme, this is given by the equation:

$$PEF_{fi} = \frac{\sum_{\hat{\pi} \in \hat{P}} L(\hat{\pi})}{\sum_{\pi \in P} L(\pi)}. \quad (9)$$

For the failure dependent scheme, we first find the mean rerouted path length for an individual path, denoted \bar{n}_π^r , given by:

$$\bar{n}_\pi^r = \frac{\sum_{e_f \in \pi} L(\hat{\pi}_d^f)}{L(\pi)}. \quad (10)$$

Now, averaging over all paths in the network provides the mean rerouted path length, which, when divided by the mean working path length gives the *PEF* for the failure dependent rerouting as:

$$PEF_{fd} = \frac{\sum_{\pi \in P} \bar{n}_{\pi}^r}{\sum_{\pi \in P} L(\pi)}. \quad (11)$$

3.2.4 Mean Memory Requirements (*MMR*)

The memory requirements for pre-stored data are quite distinct for each approach to rerouting. With the failure dependent technique, VPIs and link IDs are associated with specific failures. The information is stored in a database and is only loaded into the active VP routing tables when the crossconnect is notified of the failure. Such an operation is carried out at all the participating nodes of an alternate route detour. In contrast to this, the failure independent approach involves pre-loading translation tables of all downstream nodes with the VPIs of the alternate route; at such nodes, there are *no* database memory requirements for VPI/link ID information. This is because the translation table itself contains the mapping between input and output VPIs. Since such tables will be designed for the maximum possible number of VPs passing through a node, there is effectively no overhead. It need only be at the VP endpoints that alternate VP routing data be stored at a database, which is used to re-load the translation tables when these nodes learn that the VP has failed (Veitch et al, 1995a). Some simple assumptions will now be made to enable an approximate enumeration of memory requirements for the two rerouting paradigms. For the failure independent method, an alternate (VPI(out)/Link(out), VPI(in)/Link(in)) pairing is associated with a bidirectional VP at each endpoint, as depicted in Figure 6(a).

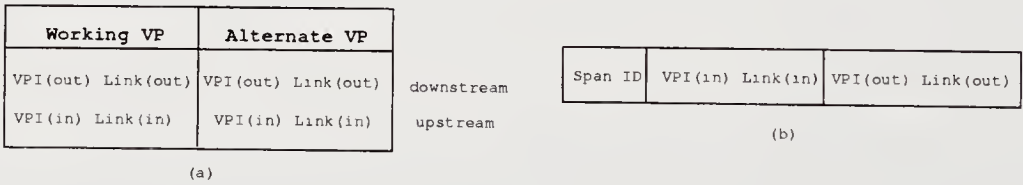


Figure 6: Storage format for alternate routing information: (a) failure independent (b) failure dependent

This covers the upstream and downstream parts of the VP. We assume that each entry takes 16 (2×8) bytes of memory for storage. If one bidirectional span disjoint protection path is allocated to each of k bidirectional VPs in a network, the total memory requirement is simply $2 \times k \times 16$ bytes. Thus, the *MMR* metric which gives the average memory requirement per node is simply:

$$MMR_n = \frac{32 \cdot k}{n}. \quad (12)$$

For failure dependent techniques, the input and output VP information corresponding to a particular span failure for one direction of a VP, is shown in Figure 6(b). It is assumed that 10 bytes are consumed with this format. If $L(\hat{\pi}_d^f)$ span hops are used in a certain protection detour, information storage is required at $L(\hat{\pi}_d^f) + 1$ nodes. Thus, in any failure dependent scheme, the memory required for a bidirectional alternate route employed when a specific span fails is:

$$2 \times (L(\hat{\pi}_d^f) + 1) \times 10 \text{ bytes.}$$

To compute the total memory requirements for a network, the above quantity will be summed over all possible span failures related to all bidirectional VPs. The *MMR* is then found by dividing by n , the number of nodes, to give:

$$MMR_{fd} = \frac{20 \times (\sum_{\pi \in P} \sum_{e_f \in \pi} (L(\hat{\pi}_d^f) + 1))}{n}. \quad (13)$$

3.2.5 Routing Computational Effort (RCE)

The computation required to produce alternate routes is non-trivial since working VP configurations may be subject to capacity and/or routing re-allocation (Sato et al, 1990) at regular intervals. This implies that protection plans have to be revised in accordance with the new VP arrangement. Fast computation is thus essential to minimise the probability that a failure will occur between the time of the working VP rearrangement and the assignment of new protection routes. We express the RCE metric in the simplest possible way, that is by the number of rerouting computations for the required protection condition, assumed throughout to be single span failures. For the failure independent scheme, since there is a protection path for each of the k working paths, we have:

$$RCE_{fi} = k. \quad (14)$$

For the failure dependent scheme, alternate routes are found for each failed path of every failed span. The total number of alternate routes required can hence be found by summing the number of spans used in each path to obtain:

$$RCE_{fd} = \sum_{\pi \in P} L(\pi). \quad (15)$$

3.3 Numerical results

A computer program was written which takes any network topology description as its input, and produces the above metrics as its output by realising each of the alternate

routing strategies. Shortest path routes were found for working paths with a random choice between equal length paths. For simplicity, all working VPs were assumed to be of unit capacity. The *SCR*, *MVR*, *PEF*, *MMR* and *RCE* metrics were computed for four grid networks of 6, 9, 12 and 20 nodes. Because of the random outcome of the shortest path algorithm, the mean result from 5 replicated computations was derived. Figures 7 and 8 display the *SCR* and *MVR* results, respectively. In all graphs, plotted points are joined up for visual convenience.

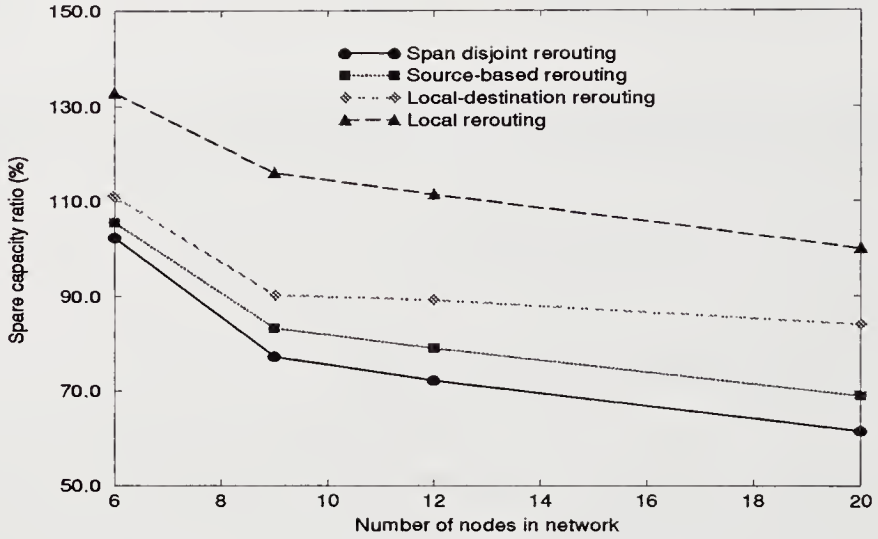


Figure 7: Spare Capacity Ratio (*SCR*) versus network size

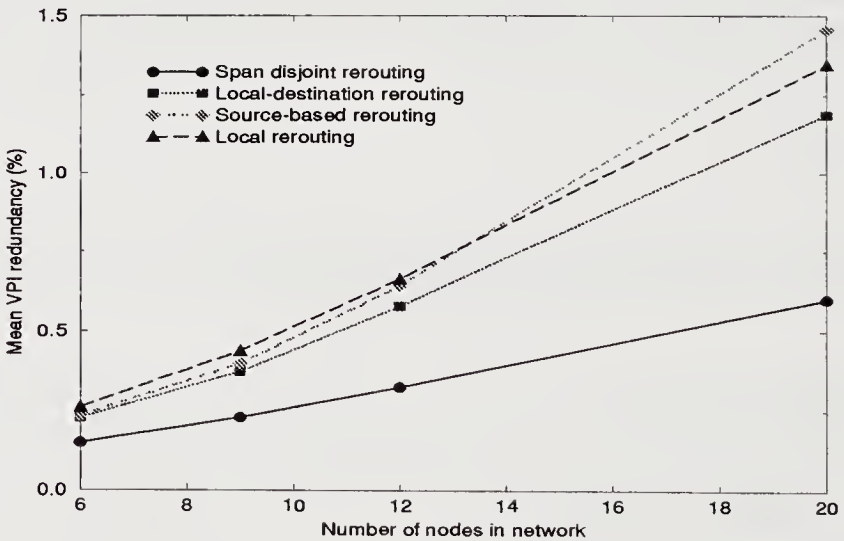


Figure 8: Mean VPI Redundancy (*MVR*) versus network size

In Figure 7, it can be seen that the failure independent i.e. span disjoint, rerouting scheme produces the lowest spare capacity ratio. As to why this is better than the source-based rerouting, it could be argued that by adopting the constraint of disjointness, backup routes are *forced* to spread the demand for spare capacity around the network. In the source-based rerouting meanwhile, the protection routes may often re-use original VP links, thus concentrating spare capacity requirements on links closer to the failure as in the local-destination approach. Of the three failure dependent approaches meanwhile, the source-based rerouting method requires the least spare capacity than the others which is thanks to the greater degree of freedom in route selection. The local-destination rerouting improves the efficiency of alternate routing design over local rerouting due to the elimination of backhauling.

From Figure 8, the VPI redundancy is greater for the failure dependent approaches, and the divergence between these and the failure independent scheme increases with network size. The main reason for this is that because different routes are allocated to individual failures which may affect a given VP, many more links are potentially involved in the rerouting process. Although the failure independent scheme used less VPIs than all of the failure dependent methods in the examples considered, this need not always be the case. One of the reasons that less than 100% spare capacity is needed for failure protection in a mesh network, is that sharing of resources between possible failures (equation (4)) is exploited. This sharing of resources between disparate failure events may be applied to VPIs. In the computations so far, a different VPI is employed for each span of every protection route, regardless of whether or not they correspond to different failures. As with capacity sharing however, the same VPI may be re-used across different failures. This is feasible in the failure dependent rerouting schemes since VPIs are stored in databases, only to be loaded into lookup tables upon failure notification. The prospect of "VPI sharing" presents an advantage of failure dependent over failure independent rerouting. This is because it is not feasible to have VPIs shared amongst protection paths defined by active VPI entries in lookup tables, since ambiguous routing would accrue.

We revise the *MVR* for the failure dependent case by defining an integer I_e^f to be the number of VPIs needed on edge e due to failure of edge e_f . The worst-case quantity of VPIs required on an edge e , is thus:

$$I_e = \max \{I_e^1, I_e^2, \dots, I_e^m\} \quad (16)$$

Hence, the total number of reserved VPIs will be:

$$N_{fd}^v = \sum_{e \in E} I_e \quad (17)$$

Which can be used in equation (8) to provide the *MVR*. The *MVR* metric was subsequently recomputed with VPI sharing allowed in the failure dependent schemes, and as shown in Figure 9, the result is a lower mean redundancy of VPIs than the span disjoint

rerouting technique. Where VPI numbers are re-used between different failures, the relative order of failure dependent schemes in terms of increasing resource demand is the same as that for the SCR metric.

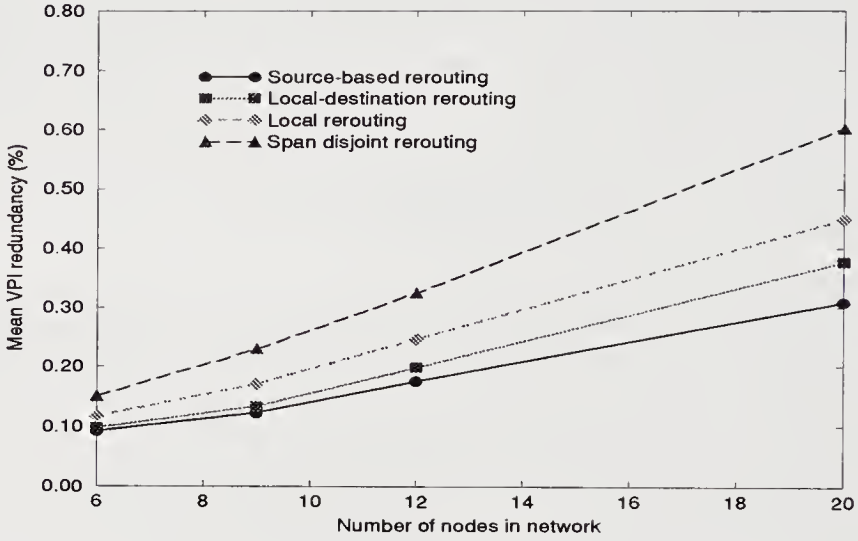


Figure 9: Recomputed *MVR* with VPI sharing, versus network size

The *PEF* metric is shown in Figure 10.

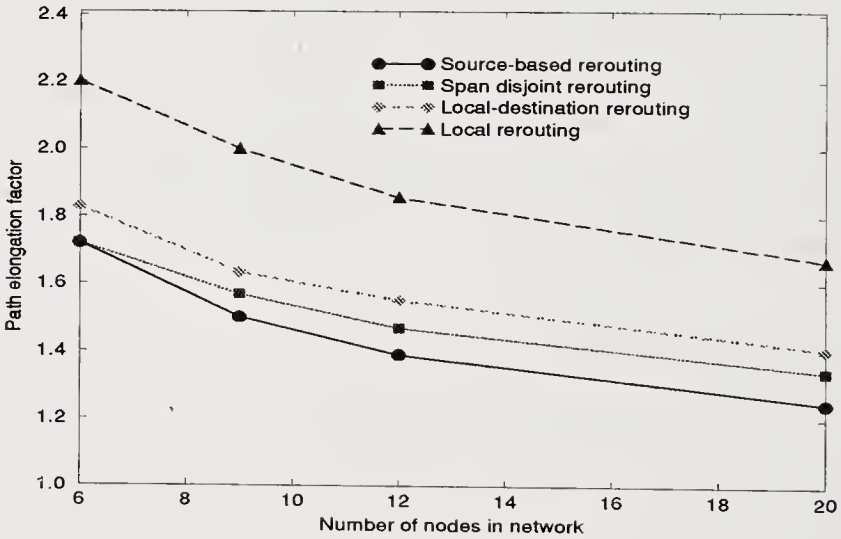


Figure 10: Path Elongation Factor (*PEF*) versus network size

From Figure 10, the minimal path elongation effects are evident with source-based rerouting, improving over the span disjoint rerouting results. The local-destination is sizeably

better than local rerouting with the latter demonstrating greatest sensitivity to path elongation, mainly due to backhauling effects. To help understand why source-based rerouting should outperform span disjoint rerouting when both techniques reroute from path terminating nodes, consider Figure 11.

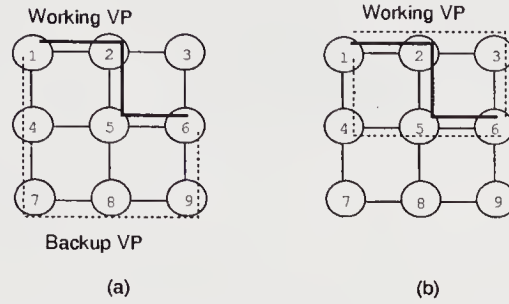


Figure 11: Potential path elongation with span disjoint rerouting

In part (a) of the Figure, the working path 1-2-5-6 is shown to be protected by backup route 1-4-7-8-9 with the failure independent rerouting scheme. There is thus a difference of 2 hops between working and protection routes. Referring to part (b) of the diagram, with the source-based rerouting version of failure dependent protection, route 1-4-5-6 could be selected for the failure of spans 1-2 or 2-5. For the failure of span 5-6 meanwhile, the new route could be 1-2-3-6. In all such cases, the working and protection routes are the same length, i.e. there is no elongation. The inferior PEF of failure independent rerouting is thus due to the disjoint criterion. It should be pointed out however, that with a more careful selection of working path route between the same nodes in Figure 11, eg 1-2-3-6, a span disjoint backup path 1-4-5-6 could be allocated yielding no elongation. This demonstrates the inherent dependence of protection routing design on the particular layout of working path routes, a point noted by Coan et al who suggested joint optimisation of working and protection layouts to achieve a truly global optimal design (Coan et al, 1991) .

The remaining metrics, *MMR* and *RCE*, are shown in Figures 12 and 13, respectively. The estimate of database storage required per node shown in Figure 12, clearly indicates the deficit between failure dependent and failure independent rerouting strategies. Indeed, the deficit enlarges with the scale of the network, whereby source-based rerouting proves to be increasingly sensitive. Of course, it may be argued that a few kilobytes of memory is unimportant, however the estimates could be misleading. The reason for this is that a mean demand per node was computed, which is fairly artificial as some maximum value would be used in practice. Also, the storage would have to accommodate future physical and logical growth, since the assumption of a single VP between each node pair will often be unrealistic.

The computational effort needed to produce rerouting information is shown in Figure 13 with no distinction between the failure dependent schemes since only the number

of required alternate routes was evaluated. If desired, suitable weighting of each scheme could allow individual curves to be fashioned, although this is not considered in this paper. The curve is striking as it highlights the sizeable computational overhead associated with failure dependent rerouting in contrast with its failure independent counterpart.

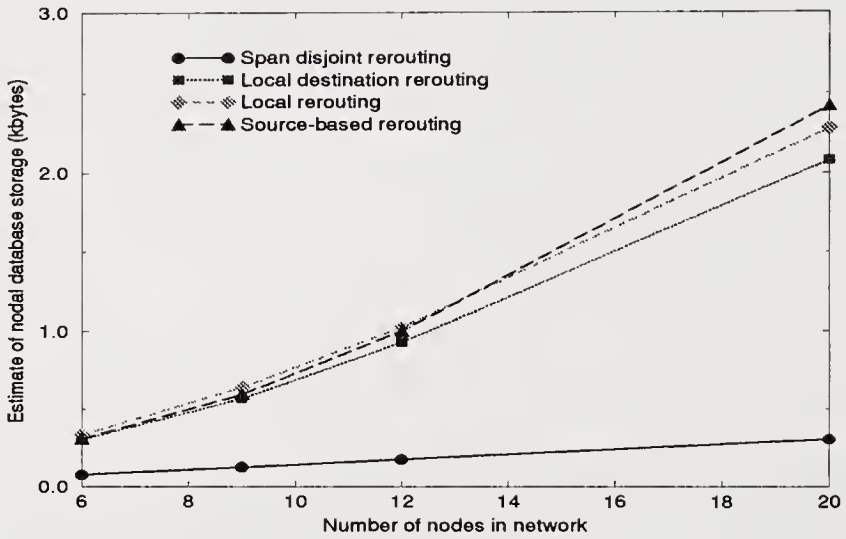


Figure 12: Mean Memory Requirements (MMR) versus network size

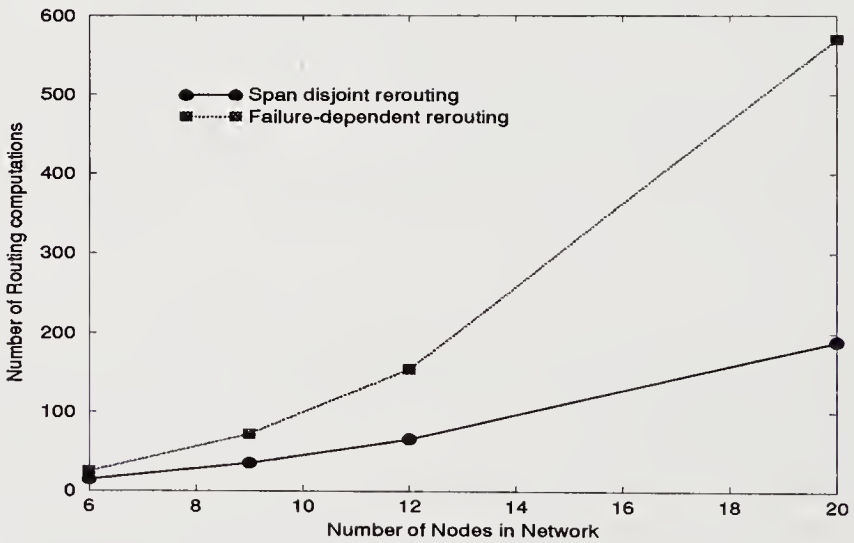


Figure 13: Routing Computational Effort (RCE) versus network size

4 DISCUSSION

4.1 Signalling protocol complexity

The preceding section presented a comparison of four pre-planned restoration schemes incorporating distinct rerouting policies, three of which are special cases of failure dependent rerouting, whilst the fourth constitutes the failure independent policy. An important facet of restoration which has not been discussed as yet, is the signalling protocol employed to activate pre-assigned routes. First, the speed with which restoration can be accomplished is paramount, since it governs the extent to which services will be adversely affected by the failure. In the AT&T paper (Anderson et al, 1994), no computer simulations of the signalling protocol to execute failure dependent restoration are described. Rather, an estimation of the restoration completion time for a 40 node network with modest processing time assumptions is cited as 58 msec. In (Veitch et al, 1995a), simulations of distributed protocols to realise failure independent backup path restoration, suggest that completion times of tens of milliseconds are possible. It can hence be postulated that comparable restoration completion times accrue with both methods of VP rerouting. Of additional concern is the ease with which protocols can be implemented. In the failure independent scheme, bidirectional F4 Operations, Administration and Maintenance (OAM) flows can be used to convey alarm and confirmation signals between the endpoints of the failed VP and the protection VP, respectively. This is a significant advantage given that certain OAM cells are already standardised (ITU-T, 1993b). With the failure dependent schemes, inter-nodal signalling channels, the properties of which have yet to be elucidated, must be employed to broadcast failure notification signals.

4.2 Planning adaptability and protocol robustness

The final issue to be considered as a basis for comparing the failure independent and failure dependent schemes, with these latter grouped as a whole, is that of robustness. First, we could analyse failure adaptability and question how each restoration scheme, in planning and execution, handles multiple span or node failures. With failure independent rerouting, transit node failures can be protected by allocating node disjoint protection paths with a suitable spare capacity allocation to match (Kawamura et al, 1994). No change to the signalling protocol is necessary with protection route activation performed in the same way as that for span failures, and no additional storage overheads are incurred. Unavailability of a protection route due to a multiple failure is easily identified with explicit confirmation of backup paths orchestrated from the endpoints (Veitch et al, 1995a). This could lead to a dynamic route searching protocol being invoked, or direct notification of the problem to a central controller. Planning for multiple span or node failures with failure dependent rerouting significantly impacts on the complexity of the whole approach. First, concerning the required planning effort, storage overheads and routing computation would increase sizeably due to the association of alternate routes with specific failures. This intractability would become accentuated with larger network

topologies. Secondly, the signalling protocol would have to be modified so that nodes which receive broadcast messages glean an unambiguous picture of the current physical network topology.

The last matter of uncertainty which puts the alternative schemes to the test is the prospective lack of spare capacity in the network with which to support rerouted traffic. Although planning is performed in conjunction with spare capacity placement, or indeed in adherence with spare capacity constraints (Veitch et al, 1995c), occasions can arise where the supply will not meet the demand. If protection routes are activated under such circumstances, the quality of service of existing connections which share common buffer and transmission resources, and are unaffected by failure in the first place, could be unacceptably degraded. Because failure independent rerouting involves explicit confirmation of protection path capacity availability (Kawamura et al, 1994, Veitch et al, 1995a), if a path cannot be supported, the situation is quickly recognised and appropriate action taken. The problem with the failure dependent approach is that there is no notion of "capacity capturing" during crossconnect table activation, which is executed for a bundle of rerouted paths at any one time. This places a question mark over the supposed robustness of failure dependent rerouting.

5 SUMMARY AND CONCLUSIONS

This paper has highlighted the fundamental differences between two pre-planned VP restoration paradigms, the failure dependent and the failure independent methods. The choice of strategy influences implementation costs in terms of spare capacity, reserved VPIs, computational overheads and memory for rerouting information storage. Furthermore, the anticipated path elongation which impacts on the delay performance experienced by rerouted connections, must be accounted for. Metrics corresponding to all these factors were formulated, then, for a variety of grid network models, a comparative evaluation was carried out between the failure independent span disjoint rerouting scheme and three distinct failure dependent rerouting policies.

The span disjoint scheme required the least spare capacity for all networks considered, with the source-based rerouting version of failure dependent restoration a close second. The important point to note is that these results were not optimised, rather, a shortest hop routing algorithm was used throughout for comparative purposes. If optimisation was performed with the minimisation of a cost function based on spare capacity, an intuitive argument would suggest that the source-based failure dependent rerouting would require less spare capacity than the failure independent case. This is due to the tailoring of alternate routes to the actual failure, something which failure independent rerouting does not cater for. The other two failure dependent schemes, local-destination and local rerouting, displayed greater demand for spare capacity, with the latter being the "greediest", due to the frequent occurrence of *backhauling* meaning the same span is re-used in a route. Regarding VPI redundancy for protection routing, the outcome depends on whether or not VPI sharing is administered in the instances of failure dependent restoration. Without

VPI sharing, the failure independent scheme requires less idle VPIs than all the failure dependent methods, otherwise, it is the failure independent method that incurs the greatest redundancy. The degree of path elongation is minimised with source-based rerouting, whilst the span disjoint scheme improves over the other two failure dependent policies. The reason for the span disjoint scheme's inferiority to the source-based method in terms of path elongation, is that certain choices of working path routes forces the disjoint backup path to use a greater number of spans than is theoretically necessary. As expected, local rerouting was the most sensitive to path elongation effects, again due to backhauling. In terms of storage overheads and routing computational effort meanwhile, failure independent rerouting exhibits a clear advantage over all failure dependent schemes, with significantly less memory required and a computational effort which is proportional to the number of paths in the network only.

It is evident that in terms of required spare capacity, the number of spare VPIs, and the anticipated elongation of paths, the two most attractive solutions to pre-planned VP restoration appear to be the failure independent scheme and the source-based rerouting version of failure dependent protection. This latter should accomplish the lowest spare capacity provisioning if optimisation is performed, and furthermore, a smaller number of VPI numbers are idled. Also, for the network models considered, the path elongation was minimised with source-based rerouting. Regarding VPI redundancy, Kawamura postulated that the ratio of working to backup VPs in any link does not cause concern for VPI availability where disjoint backup paths are assigned (Kawamura et al, 1994). Although span disjoint rerouting demonstrated greater sensitivity to path elongation, this could be remedied by exercising a joint working/protection VP layout which minimises elongation effects. On the foundation of these observations therefore, it may be argued that the principal advantage of source-based rerouting is the prospect of spare capacity minimisation, though the computational effort needed to attain this, and how much gain over the failure independent scheme would accrue, remains open for investigation.

The potential advantage of source-based rerouting is offset by the distinct disadvantage of much greater storage overheads needed to support alternate routing plans. In addition, the routing computation will be far more intense for all failure dependent techniques compared with the failure independent approach. This combination of factors tends to swing in favour of the failure independent protection routing paradigm. This preference is consolidated by analysis of the qualitative issues related to protocol complexity and robustness. It was discussed in the penultimate section of the paper how backup path activation could be executed with simple OAM cell transmission protocols. These same protocols could be used whether a span or nodes fail. Indeed, to plan for node failures, node disjoint backup routes can be allocated, with no additional storage overheads incurred to support this mode of failure recovery. Furthermore, the confirmation of backup path availability allows detection of multiple failure or limited spare capacity conditions. All of these features of failure independent rerouting are in sharp contrast to the failure dependent approach which requires significant extra computation and storage space to accommodate other failures besides single span. The restoration protocol itself is complicated by unanticipated failures, and if there is limited spare capacity to support rerouted paths, there is no specified distributed mechanism to recognise the syndrome.

To conclude, the failure independent rerouting scheme for pre-planned Virtual Path restoration incorporates properties of resource efficiency, low implementation complexity and robustness, which combine to make it a suitable foundation for planning survivable ATM networks.

References

- Anderson J., Doshi, B.T., Dravida, S. and Harshavardhana, P. (1994). Fast Restoration of ATM Networks. *IEEE Journal on Selected Areas in Communications*, January, 128-138.
- Coan, B.A., Leland, W.E., Vecchi, M.P., Weinrib, A. and Wu, L.T. (1991). Using Distributed Topology Update and Preplanned Configurations to Achieve Trunk Network Survivability. *IEEE Transactions on Reliability*, October, 404-416.
- ITU-T Recommendation I.311 (1993a). B-ISDN General Network Aspects.
- ITU-T Recommendation I.610 (1993b). B-ISDN Operation and Maintenance Principles and Functions.
- Kawamura, R., Sato, K-I. and Tokizawa, I. (1994). Self-Healing ATM Networks Based on Virtual Path Concept. *IEEE Journal on Selected Areas in Communications*, January, 120-127.
- Sato, K-I., Ohta, S. and Tokizawa, I (1990). Broadband ATM Network Architecture Based on Virtual Paths. *IEEE Transactions on Communications*, August, 1212-1222.
- Sosnosky, J. (1994). Service Applications for SONET DCS Distributed Restoration. *IEEE Journal on Selected Areas in Communications*, January, 59-68.
- Veitch, P.A., Smith, D.G. and Hawker, I. (1995a). A Distributed Protocol for Fast and Robust Virtual Path Restoration. *Proc. 12th IEE UK Teletraffic Symposium*, Windsor, England.
- Veitch, P.A., Smith, D.G. and Hawker, I. (1995b). Restoration Strategies for Future Networks. *Electronics & Communication Engineering Journal*, June, 97-104.
- Veitch, P.A., Smith, D.G. and Hawker, I. (1995c). The Design of Survivable ATM Networks, in *Performance Modelling and Evaluation of ATM Networks, Volume 1* (Ed. D.D. Kouvatsos), 517-534. Chapman & Hall, London.
- Wu, T-H. (1992). *Fiber Network Service Survivability*. Artech House.

Biographies

Paul Veitch is currently studying for a PhD degree in Electronic & Electrical Engineering at the University of Strathclyde, where, in 1993, he obtained his MEng degree with distinction in the same discipline. The research into ATM restoration techniques is co-sponsored by the EPSRC and BT Labs as part of a CASE award.

Geoffrey Smith is professor of communications at the University of Strathclyde. His research interests include management and control of broadband networks.

Dr Ian Hawker leads a team at BT research laboratories whose interests lie in modelling and performance evaluation of future networks.

Virtual Path Bandwidth Control Versus Dynamic Routing Control

I. Z. Papanikos, M. Logothetis and G. Kokkinakis
*Wire Communications Laboratory,
Dept. of Electrical and Computer Engineering,
University of Patras,
261 10 Patras, Greece.
Tel. +30 61 991722 Fax: +30 61 991855
E-mail: m-logo@wcl.ee.upatras.gr*

Abstract

Virtual Path Bandwidth (VPB) control and Virtual Circuit Routing (VCR) control are competitive control schemes for traffic management in ATM networks. The objective of both controls is to minimize the Call Blocking Probability (CBP) of the congested end-to-end links, under constraints posed by the transmission links capacity of the network. Firstly, we compare the performance of two VCR control schemes, the **DAR** and **DCR**, well-known in the environment of STM networks, considering several trunk reservation parameters and different control intervals. Secondly, we compare the performance of VPB control schemes with that of VCR control schemes, both under static and dynamic traffic conditions. Under static traffic conditions the efficiency of the two control schemes in minimizing the worst CBP of the network is examined, whereas under dynamic traffic conditions their response time is measured by means of simulation. In short, VPB control is more effective than VCR control when the traffic fluctuation is large while VCR control has a faster response time than VPB control.

Keywords

Virtual Path Bandwidth Control, Dynamic Routing Control, ATM networks.

1 INTRODUCTION

In ATM networks, network/traffic management has a layered structure of two levels, the Call-level and the Cell-level, which correspond to the distinction of traffic in call and cell components, respectively. We concentrate on the Call-level traffic management and especially on controls which drastically influence the global performance of an ATM network under constraints posed by the bandwidth capacity of transmission links. Virtual Path Bandwidth (VPB) control and Virtual Circuit Routing (VCR) control are the main controls strongly related to the transmission links capacity. Their performance is evaluated by the Call Blocking Probability (CBP). Bandwidth and trunk reservation controls are also related to the transmission links capacity and closely cooperate either with VPB or VCR control.

In this paper, we compare the performance of VCR control schemes, also called Dynamic Routing (**DR**) (Mase, 1989), with the performance of VPB control schemes (Logothetis 1992, Shioda 1994), in the environment of ATM networks.

The VCR control objective is to provide an alternate route for each Virtual Circuit Connection (VCC) that fails to be established on the first choice (direct) Virtual Path Connection (VPC), exploiting the spare capacity of the network. The VPB control objective is to rearrange the installed bandwidth of the VPs according to the offered traffic fluctuation so as to minimize the worst (maximum) CBP of all end-to-end links.

Several DR control schemes have been proposed for use in the traditional telephone networks:

- a) Dynamic Non-Hierarchical Routing (DNHR), a time-dependent routing scheme developed by AT & T (Ash, 1990),
- b) Trunk Status Map Routing (TSMR) (an extension of DNHR) that modifies the routing patterns calculated by DNHR considering the trunk status (Ash, 1985),
- c) Dynamic Alternative Routing (DAR), a decentralized state-dependent routing developed by British Telecom (Stacey 1987, Key 1990),
- d) Dynamically Controlled Routing (DCR), a centralized version of the state-dependent dynamic routing (Rengier 1983, Cameron 1983),
- e) State and Time-dependent Routing (STR), a hybrid routing scheme that combines the time-dependent control at the routing pattern definition and state-dependent control at the VC-level routing definition, proposed by NTT (Mase, 1990).

We have chosen two of the above DR control schemes to be considered as VCR control schemes in ATM networks: the decentralized control scheme DAR and the centralized control scheme DCR. Before comparing their performance with that of VPB control schemes, their performance in minimizing the worst CBP is comparatively examined, when they cooperate with several Trunk Reservation control schemes, or when different control intervals are considered.

The performance of the VCR and VPB control schemes is examined under static and dynamic traffic conditions on a test-bed ATM-network of 10 nodes, in a ring topology, accommodating two service-classes. Under static traffic conditions we examine the performance of VPB and VCR controls in minimizing the worst CBP of the whole network. The applied VPB control is optimal and is obtained analytically, through a global network optimization model. The results of the application of the VCR control schemes are obtained through simulation. Under dynamic traffic conditions, we examine the response time of the above control schemes. For the application of VPB control we consider the Medium-Term VPB control scheme, described in

reference (Logothetis, Shioda, 1995), with a control interval long enough, because the required time for bandwidth rearrangement is considerably long, due to the existing call connections at that time-point. As far as the incorporated bandwidth and trunk reservation control schemes are concerned, the bandwidth reservation scheme which equalizes the CBP of the two service-classes is considered for the VPB control, while several trunk reservation schemes are considered for the VCR control schemes. Concerning the dynamic traffic condition, we consider that traffic fluctuates according to a step function (theoretical case), applied on one switching pair, in one traffic-flow direction only.

This paper is organized as follows: In Section 2 an ATM network architecture is described which is appropriate for the applicability of VPB and VCR control schemes. In Section 3, the objective and the VPB control schemes are presented. Section 4 includes three subsections. In subsection 4.1 and 4.2, the VCR control schemes, DAR and DCR, respectively, are described and the calculation of the involved CBP in the VPs of an ATM network is given. In subsection 4.3 the two VCR control schemes are comparatively examined. Firstly, they are compared in respect to the resultant average CBP of the network, under static traffic condition and in cooperation with several trunk reservation control schemes. Secondly, the same comparison is carried out when the best trunk reservation control scheme is considered for cooperation (obtained from the first comparison) and the control (update) interval of the DCR control scheme varies. In Section 5, the VPB and VCR control schemes are comparatively examined, under static (subsection 5.1) and dynamic (subsection 5.2) traffic conditions. As a conclusion, we summarize the results of this paper in section 6.

2. ATM NETWORK ARCHITECTURE

An ATM network architecture is considered in which each ATM switch (ATM-SW) is accompanied by an ATM Cross-Connect (ATM-XC) system. The ATM-XCs are interconnected by a ring transmission line and compose the backbone network (Figure 1a). This architecture has the advantage of simplicity and offers higher transmission line utilization (Sato, 1990). The transmission links are assumed bi-directional. A connection between two ATM-SWs is established via any available path that has been registered in a table, called Routing Table (**RT**). Under the consideration of this paper the route of a path between two ATM-SWs passes through ATM-XCs only.

Other network topologies could be also considered. In the topology of the backbone network of Figure 1a, two parts can be distinguished to make our study easier: one composed of the ATM-XCs, called outer network and another composed of the interconnected ATM-XCs, called inner network.

Thanks to the Virtual Path (VP) concept, the traffic management by reallocating the established bandwidth of the paths (VPB management) according to the traffic variations becomes favorable in ATM networks. The concept of VP, whereby two ATM-SWs face only the direct logical (imaginary) link (VP) between them, makes the structure of the backbone network transparent to the ATM-SW pairs. This is due to flexibility of the ATM-XCs to provide the required bandwidth in the end-to-end links (VP connections) of the ATM-SWs. Therefore, from the VPB management point of view, the whole ATM network is equivalent to a meshed network in which only the direct links are used (Figure 1b). These links represent the

VPCs.

Since we assume the equivalent mesh network architecture, where the ATM-SWs are fully interconnected with VPCs, the first choice route for a VC to establish a VCC, is its direct VPC. When the VCC is blocked at the first choice route, an alternate route will be attempted (according to the applied VCR control scheme) which consists of two VPCs. This routing scheme meets the basic requirements for the application of the well-known DR control schemes of the STM networks (Yokoi, 1995).

The VCR controller can be either a decentralized controller, like the DAR, or a centralized one, like the DCR. In the case of a decentralized control scheme, in each ATM-SW there is one VCR controller who is informed about the traffic-flow condition in the VPCs of the network, by counting the number of VCC

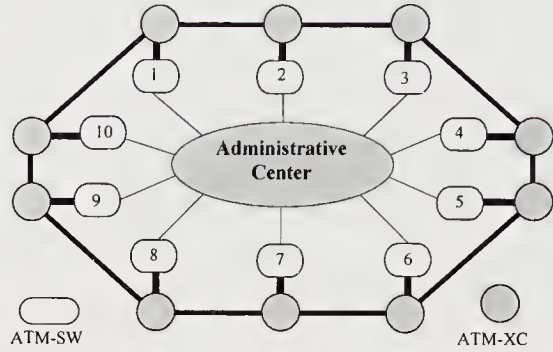


Figure 1a ATM network architecture.

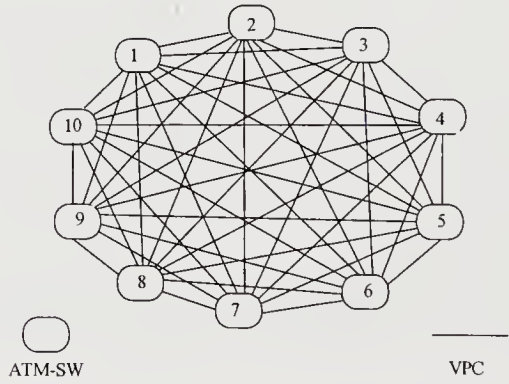


Figure 1b Equivalent meshed VPC network.

failures, in order to define the route for the next call arrival (next VCC). On the other hand, a centralized VCR controller is located at a network management centre and determines alternative VPCs to realize a VCC, for each ATM-SW pair of the network. This is done by receiving every few seconds the traffic conditions of the VPCs from each ATM-SW and exploiting the idle capacity of the VPCs.

The VPB controller is located at an administrative centre (centralized controller). It communicates with the ATM-SWs to collect the measurements of carried traffic and blocking during each control interval. Based on these measurements, it calculates the offered traffic. From the offered traffic, the installed bandwidth in the transmission links and the VPs listed in the RT, the VPB controller determines the allocation of the bandwidth to the VPs, by solving a large network optimization model. Then, it updates the data relevant to the VP bandwidth in the ATM-SWs. The realization of the produced VPB allocation is executed by the ATM-SWs simultaneously, after a delay due to the existing call-connections at the time point of bandwidth rearrangement. The ATM-SWs increase or decrease the number of cells which have a specific Virtual Path Identifier (Saito, 1991) when the bandwidth of this VP is increased or decreased, accordingly. It is worth mentioning that no communication between the VPB controller and the ATM-XCs is required.

3 VPB CONTROL

Telecommunication networks are designed to convey the traffic of all switching pairs so as to meet a pre-described QOS. Due to traffic variations from hour to hour the traffic load on some switching pairs is below the forecasted value and free bandwidth results. On the other hand, overloads occurring at the same time on other switching pairs cannot use the free bandwidth of the network, if it is not possible to transfer the surplus bandwidth towards the congested switching pairs. This is the work of VPB control. It reallocates the bandwidth of the VPs according to the offered traffic so as to improve the global performance of the network, under constraints posed by the transmission links capacities. The resultant distribution of the totally installed bandwidth to the VPs is the VPB allocation.

To rearrange the VP bandwidth dynamically, the following types of VPB control schemes have been proposed:

- a) Very-Short-Term control schemes based on the information of the concurrent connections in the VPs (Ohta, 1988), with control interval less than 5 min.
- b) Short-Term control schemes based on the blocking measurements taken during the control interval which ranges from several minutes to a few hours (Shioda, 1991).
- c) Long-Term control schemes based on traffic prediction with control interval ranging from a few hours to a few days (Monteiro, 1990).
- d) Medium-Term VPB control based on traffic measurements, with control interval ranging from several minutes to a few hours (Logothetis, Shioda, 1995).

The Very-Short-Term and the Short-Term control must be distributed control schemes in order to respond quickly to sharp traffic fluctuations and absorb them. To achieve this, they need very simple computations. They can ignore the traffic characteristics of service-classes (Ohta, 1988), which is an important advantage in the B-ISDN environment. The Very-Short-Term control achieves an optimal network performance. The implementation, however, of this control scheme is very difficult and, therefore, it is only of theoretical value. A large number of control steps is needed, especially when the traffic volume is large. The Short-Term control schemes are readily implemented but they lack optimality.

On the other hand, the Long-Term control is a centralized control where the controller aims at

an optimal network performance in the control interval by solving a large network optimization problem. However, the controller is based on the prediction of the offered traffic which is a time consuming task, though it is not possible to be accurate. Therefore, the importance of the achieved optimality is weakened. The main advantage of the Long-Term control schemes is that they can easily be implemented, because VP bandwidth is rearranged only a few times per day.

The Medium-Term VPB control scheme reconciles the advantages and disadvantages of the Short-Term and Long-Term control schemes. The controller must be a centralized one in order to optimize the network performance globally within its control interval. The control interval must be rather short in order to respond satisfactorily to medium-term traffic fluctuations. Short-term traffic fluctuations could be absorbed by the implementation of VCR control in a further stage. To achieve this Medium-Term VPB control, the controller formulates a global network optimization model which is driven from the offered traffic, determined from on-line measurements of the carried traffic and the CBP of each service-class of the network. The optimization criterion is to minimize the worst CBP of all VPCs (Logothetis 1993, Logothetis 1995).

4 VCR CONTROL

VCR control is an alternate dynamic routing method that updates the set of possible alternate VPCs for each ATM-SW pair based on the state of the network (state-dependent), or according to preplanned routing patterns calculated so as to meet the forecasted traffic demand for each time period of the day (time-dependent). Benefits of the dynamic alternate routing in comparison to the fixed alternate routing are: the higher utilization of network resources (and hence cost savings) and the tolerance against network failures and traffic fluctuations.

In this paper, two conventional dynamic routing control schemes, the Dynamic Alternative Routing and the Dynamically Controlled Routing, are examined in their applicability to ATM networks.

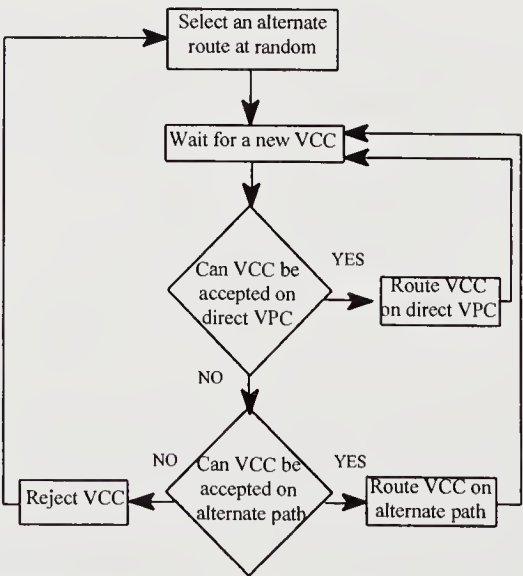


Figure 2 Flow diagram for DAR.

4.1 Dynamic Alternative Routing (DAR)

DAR is an example of a decentralized routing control scheme. According to this algorithm, a VCC that fails on the first choice VPC (direct) is offered to the current-choice alternate route (composed by two VPCs) and if it is blocked, a new current-choice is selected at random from all possible alternate routes, to be used for the next VCC attempt (Figure 2).

Performance evaluation of an ATM network controlled by DAR

To evaluate the performance of an ATM network controlled by DAR, we determine the CBP of the VPCs. For the long-run stationary behavior of the network, we extend the methodology found in references (Gibbens 1989, Key 1989, Mitra 1991) to the ATM environment, considering that each VP is commonly shared by two service-classes (c_k) with b_{ck} ($k=1,2$) required bandwidth per call.

The following notations are used:

- V_s : Bandwidth assigned to the VPC s .
- $r(1)$: First-choice VPC used by the switching pair r .
- $r(2)$: Alternate route of two VPCs used by the switching pair r .
- R_r : Set of all possible alternate routes for the switching pair r .
- R_s : Set of switching pairs that use the VPC s as a first or as a second VPC of their alternate routes ($r \in R_s, r:s \in r(2)$).
- a_r^l : Probability that the alternate route l is selected for the switching pair r .
- $l(s)$: Alternate route that contains the VPC s .
- $l_1(s)$: First VPC of the alternate route $l(s)$.
- $l_2(s)$: Second VPC of the alternate route $l(s)$.
- $\rho_{1,s}^{c_k}$: First-choice (direct), Poisson traffic offered to the VPC s , by the service-class c_k .
- $\rho_{2,s}^{c_k}$: Alternate traffic (assumed as Poisson traffic) offered to the VPC s , by the service-class c_k .
- $B_{1,s}^{c_k}$: CBP for the first-choice traffic offered to the VPC s , by the service-class c_k .
- $B_{2,s}^{c_k}$: CBP for the alternate traffic offered to the VPC s , by the service-class c_k .

The alternate traffic of each service-class c_k offered to the VPC s , is determined as:

$$r_{2,s}^{c_k} = \sum_{r:s \in r(2)} r_{1,r(1)}^{c_k} B_{1,r(1)}^{c_k} a_r^{l(s)} (1 - B_{2,s'}^{c_k}), \quad s' \in r(2) - s, \quad k=1,2 \quad (1)$$

After a long-run time, since the selection of alternate routes is uniform and the blocking rates over the two VPCs of an alternate route are equalized, the Selection Probability, $a_r^{l(s)}$, of an alternate route results to be inverse proportional to the blocking of the alternate route:

$$a_r^{l(s)} \propto \frac{1}{(1 - (1 - B_{2,l(s)})(1 - B_{2,l_2(s)}))} \quad \text{and} \quad \sum_k a_r^k = 1, \quad k \in R_r \quad (2)$$

For the determination of the CBPs of each VPC of the network, we consider only the Call-level characteristics of the service-classes. We propose the recursive formula found in references (Kaufman 1981, Roberts 1982) to be used for the determination of CBPs, taking into account the bandwidth reservation control between the service-classes. As it has been observed (Logothetis, 1992), this formula has a high accuracy especially when the service-classes have the same mean service-time. To apply this formula to the DAR, we have to consider that four traffic streams, t_k ($k=1,2,3,4$), are offered to each VPC. The traffic streams t_1 and t_3 are due to the first and the alternate offered traffic of the first service-class, respectively, whereas the t_2 and t_4 are due to the first and the alternate offered traffic of the second service-class, respectively.

The CBPs of the VPCs are determined as:

$$B_{t_k} = \frac{1}{G} \sum_{n=1}^{b_{t_k} + R(t_k) - 1} G(V_s - n) \quad (3)$$

where

$$G = \sum_{i=1}^{V_s} G(i) \quad (4)$$

$$G(i) = \frac{1}{i} \sum_{k=1}^4 r_{t_k} D_{t_k}(i - b_{t_k}) G(i - b_{t_k}) \quad \text{for } i = 1, \dots, V_s \quad (5)$$

$$D_{t_k}(i - b_{t_k}) = \begin{cases} b_{t_k} & \text{for } i \leq V_s - R(t_k) \\ 0 & \text{for } i > V_s - R(t_k) \end{cases} \quad (6)$$

$R(t_k)$ is the bandwidth reserved for each traffic stream due to the Bandwidth and the Trunk Reservation Control (Figure 3).

In this way, we have formulated in the ATM environment a system of equations (1-6) which is solved by an iterative method in the computer. This system is equivalent to the fixed-point system of equations which is valid for the STM environment.

4.2 DYNAMICALLY CONTROLLED ROUTING

DCR is an example of a centralized routing control scheme. It uses a central processor to find an alternate route (composed by two VPCs) for each switching pair of the network, based on the free capacity of the VPCs of the whole network. The central processor:

- gathers, during its control interval, all the appropriate information (VPC trunk status, traffic, etc.), from each ATM-SW,
- calculates, for each switching pair, the alternate route selection probability which is proportional to the measured idle capacity of the alternate route set,
- selects the alternate route based on the selection probability,
- sends the alternate route information to the ATM-SWs.

The control (update) interval of DCR is in the order of a few seconds, whereas the theoretical case of a zero control interval can be considered as well.

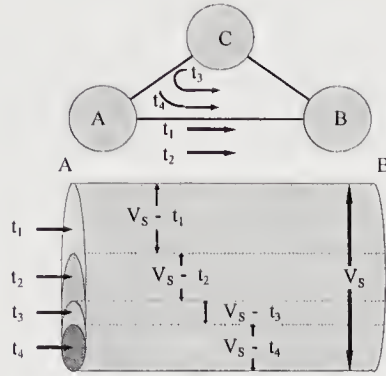


Figure 3 Bandwidth and Trunk Reservation in a VPC.

Determination of Call Blocking Probability

For the determination of CBP in an ATM network controlled by the DCR, the same notations with the DAR system are used. In addition to them the following notations are used:

- L_s : Residual capacity of the VPC s.
- C_s : Occupied bandwidth of the VPC s.
- T_s : Trunk Reservation number of the VPC s.
- L_{l1} : First VPC of the alternate route $l \in R_r$.
- L_{l2} : Second VPC of the alternate route $l \in R_r$.

The DCR control solves the same system of equations, as the DAR controller, under stationary traffic conditions (Girard, 1990). However, in the DCR, the Selection Probabilities of the alternate routes are computed, for zero update interval, as follows:

Firstly, the residual capacity L_s of VPC s, is computed as:

$$L_s = V_s - C_s - T_s \quad (7)$$

The C_s is calculated as the total traffic carried on the VPC s:

$$C_s = \sum_{k=1}^2 (r_{1,s}^{c_k} (1 - B_{1,s}^{c_k}) + r_{2,s}^{c_k} (1 - B_{2,s}^{c_k})) b_{c_k} \quad (8)$$

The residual capacity of the alternate route of two VPCs is computed as:

$$\bar{L}_l = \min(L_{l1}, L_{l2}) \quad (9)$$

and the Selection Probability of the alternate route is given as:

$$a_r^{l(s)} = \frac{\bar{L}_{l(s)}}{\sum_{k \in R_r} \bar{L}_k} \quad (10)$$

4.3 Comparison of the VCR control schemes

Two reservation parameters, the Bandwidth and Trunk Reservation numbers, must be considered for a VCR control scheme in order to improve the performance of multi-service networks, such as ATM networks. Bandwidth Reservation aims at guaranteeing the QOS of each service-class multiplexed in a VP, by reserving some fraction of the VP bandwidth for the service-classes which require larger bandwidth. So, calls of service-class c_k are refused to be connected when less than $t(c_k)$ bandwidth is available in the VP. By a proper selection of the Bandwidth Reservation number the resultant CBP of the two service-classes, in each VP, can be equalized. On the other hand, Trunk Reservation aims at guaranteeing the network stability when an alternate routing scheme is applied. It protects the first offered traffic to a VP against alternate routed traffic which makes use of this VP. It depends on VP bandwidth and traffic load offered to the VPs.

Table 1

Trunk Reservation Number	Maximum Traffic Fluctuation (%)							
	10	20	30	40	50	60	70	80
0	1.91	2.59	3.50	3.47	4.49	5.23	5.77	6.14
	1.72	1.62	2.46	3.31	4.22	4.85	5.27	5.99
24	1.07	1.12	1.43	1.64	1.98	2.39	2.86	3.31
	0.72	0.90	1.17	1.74	1.95	2.48	2.85	3.39
48	1.00	1.10	1.31	1.62	2.03	2.31	2.87	3.20
	0.86	0.97	1.34	1.44	1.81	2.17	2.59	3.13
72	1.22	1.44	1.56	1.87	2.20	2.65	3.09	3.36
	0.96	1.07	1.34	1.65	2.11	2.45	2.88	3.12

In Table I, the average CBP of an ATM network (described below) which operates with DAR (first number in Table I) or DCR (second number in Table I) VCR control schemes, versus Trunk Reservation numbers is given. The same Trunk Reservation number is considered for each VP-link. The Bandwidth Reservation number is such that the CBPs of the two service-classes are equalized. Table I shows that in case of small traffic fluctuation the CBP of the network increases as the Trunk Reservation number increases. In case of large traffic fluctuation a larger Trunk Reservation number is needed.

The performance of the two VCR control schemes described above, is examined in the ATM network of 10 ATM-SWs (see below). The Trunk Reservation number is taken from Table I and corresponds to the best one for each traffic fluctuation. Five versions of the DCR are presented. The DCR-0 with zero update interval and the DCR-5, DCR-10, DCR 15, DCR-20 of update interval 5, 10, 15, 20 sec, respectively. Figure 4, shows the average CBP of the whole network operating with the DAR control or the DCR-0, DCR-5, DCR-10, DCR-15, and DCR-20 control schemes versus traffic fluctuations. The results show that the DCR-0 has the best performance, while the performance of DAR is better than the DCR-5, DCR-10, DCR-15 and

DCR-20. In practice, however, the DCR-0 cannot be applied; since this control is a centralized one, a control interval of the order of a few seconds is required, at least. Figure 5 shows the average CBP of the whole network operating with DAR and DCR versus control interval. When the control interval is small the performance of the DCR is better than that of the DAR.

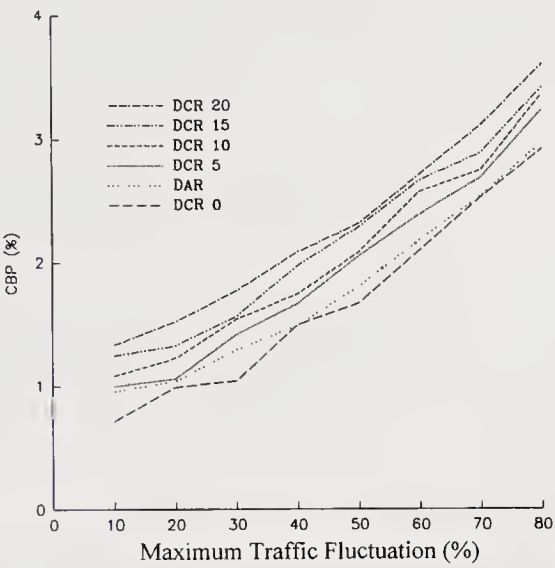


Figure 4 Average CBP versus maximum traffic fluctuation for DAR and DCR.

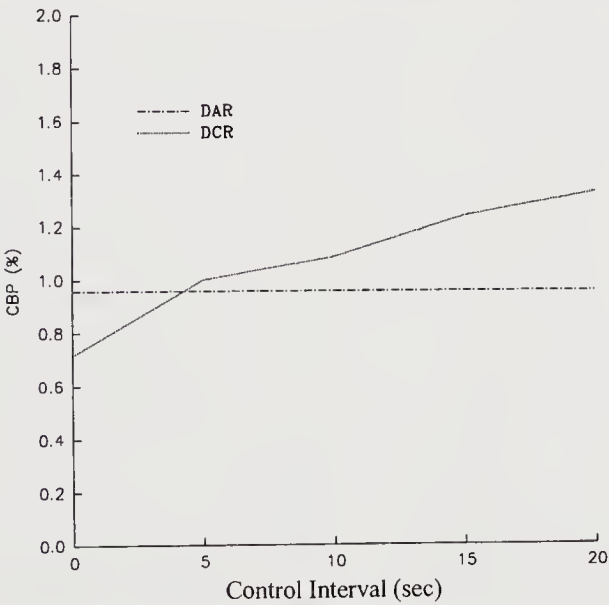


Figure 5 Average CBP versus control interval for DAR and DCR.

5 COMPARISON OF VPB CONTROL WITH VCR CONTROL

The performance of the VPB control and the VCR control are compared under static and dynamic traffic conditions on a 10 ATM-SWs ring ATM network. Under static traffic conditions the average and the worst CBP of the network are presented. Under dynamic traffic conditions, the response time of the two traffic controls is examined.

Two service-classes are accommodated in the network. The required bandwidth per VCC for the first service-class is 64 kbps (considered as bandwidth unit or one trunk capacity), and for the second service-class is 1.536 Mbps (i.e. 24 bandwidth units). Because of the Bandwidth Reservation Control, 1.472 Mbps (23 bandwidth units) are reserved to benefit the second service-class, in each VP. The Trunk Reservation number is taken from Table I and corresponds to the best one for each traffic fluctuation. Both service-classes have exponentially distributed holding times with mean value of 100 sec.

The VPs of the network are dimensioned so as to satisfy the grade-of-service of 3% (CBP). The traffic offered to each ATM-SW are 260 Erl and 12 Erl for the first and the second service-class, respectively. The VP bandwidth is 43.008 Mbps (672 bandwidth units), for each VP. The bandwidth of a transmission link (between two ATM-SWs) is calculated as the sum of the VPs that use this transmission link.

5.1 Static Traffic Conditions

The average and the worst CBP of the whole network are examined when the offered traffic fluctuates randomly according to the uniform distribution by a maximum of 10% of the design traffic-load, reaching to 80% in steps of 10%.

In Figure 6, the worst CBP of the network is shown versus the maximum traffic fluctuation, when VPB control, VCR control and No-Control are applied to the network. Figure 6a shows the results of No-Control, DAR and VPB control comparatively, whereas Figure 6b shows the results of No-Control, DCR and VPB control. The resultant worst CBP of DAR and DCR is obtained through simulation (Logothetis, Kokkinakis, 1995), while the results of VPB control are obtained analytically and are optimal (Logothetis 1993, Logothetis 1995). As we can observe, the VPB control is more effective when the maximum traffic fluctuation is large, while when the traffic fluctuation is small the VCR controls perform better than VPB control.

In Figure 7, the average CBP of the whole network is shown versus the traffic fluctuations. Figure 7a presents the results of No-Control, DCR and VPB control comparatively, whereas Figure 7b presents the results of No-Control, DAR and VPB control. When a VCR control is applied, the network performance in respect to the average CBP is better for all traffic fluctuations. It is worth mentioning that the objective of VPB control is to minimize the maximum CBP of the network; therefore, when this criterion is satisfied, no action is taken in order for the average CBP of the network to be improved.

Figure 8a and 8b comparatively show the worst CBP and the average CBP of the network versus the maximum traffic fluctuation, respectively. The DAR and DCR curves of Figure 6a and 6b are portrayed together in Figure 8a. Likewise, the DCR and DAR curves of Figure 7a and 7b are portrayed together in Figure 8b.

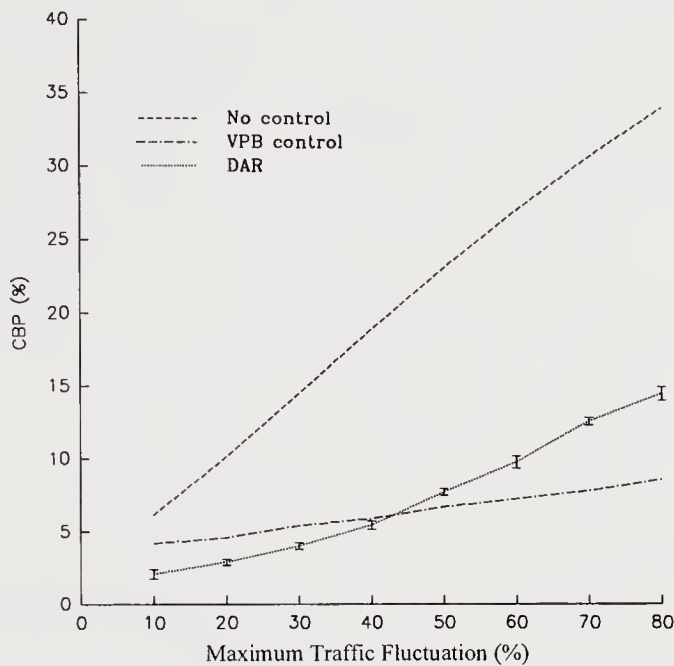


Figure 6a Worst CBP versus maximum traffic fluctuation for the VPB and DAR controls.

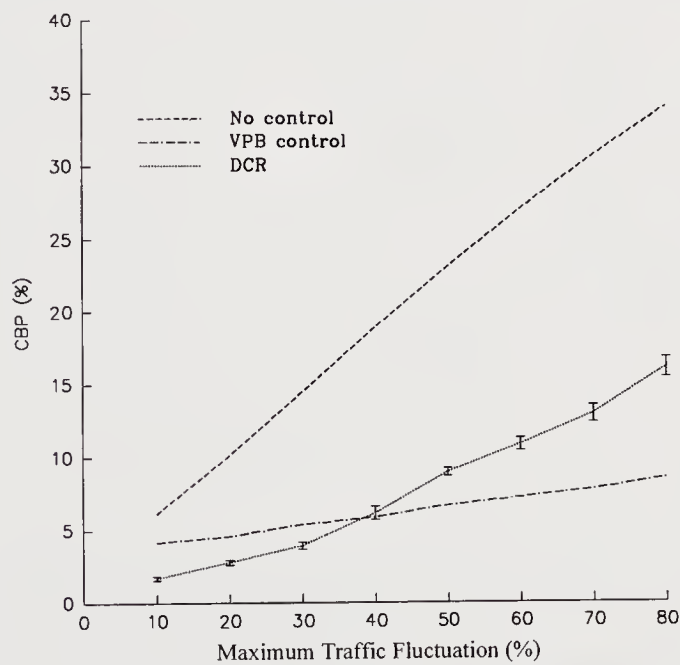


Figure 6b Worst CBP versus maximum traffic fluctuation for the VPB and DAR controls.

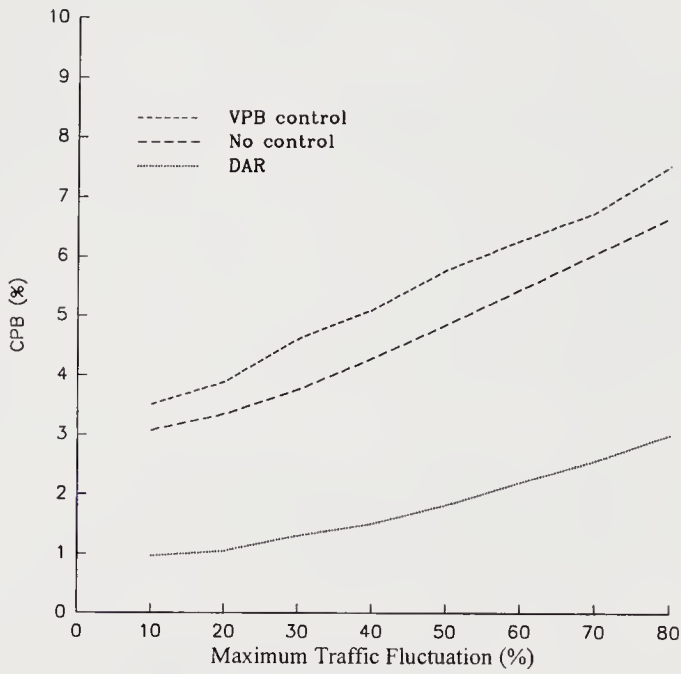


Figure 7a Average CBP versus maximum traffic fluctuation for the VBP and DAR controls.

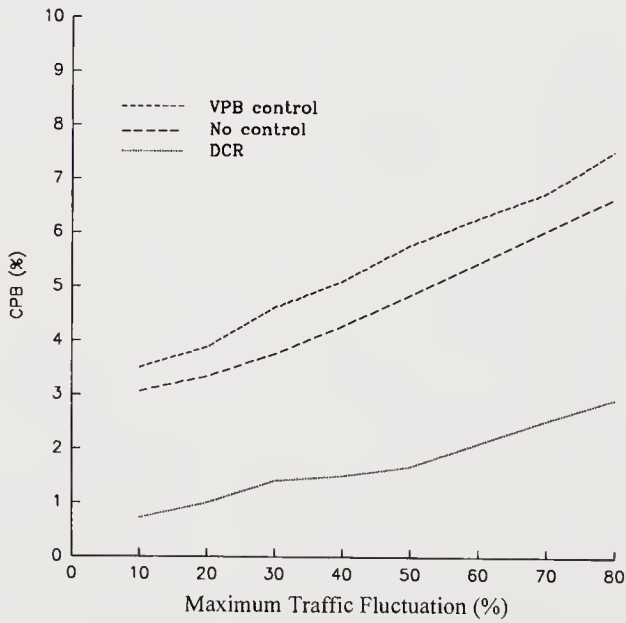


Figure 7b Average CBP versus maximum traffic fluctuation for the VBP and DCR controls.

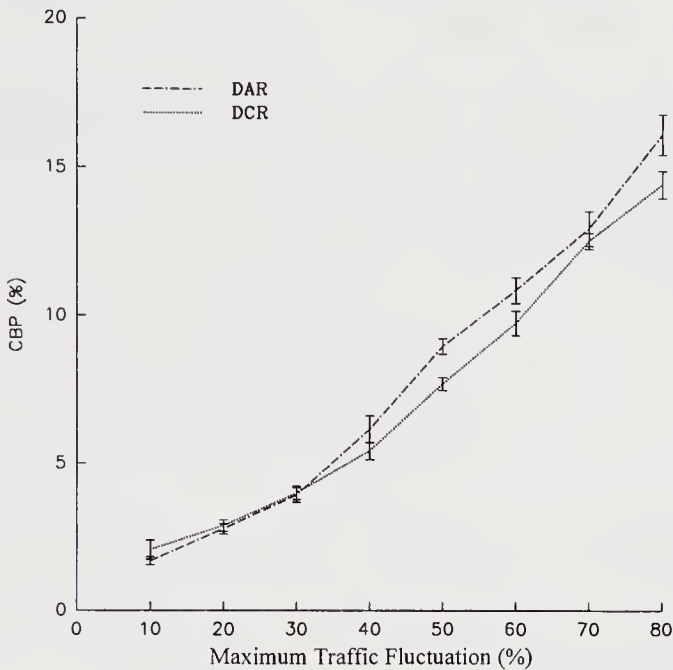


Figure 8a Worst CBP versus maximum traffic fluctuation for the DAR and DCR controls.

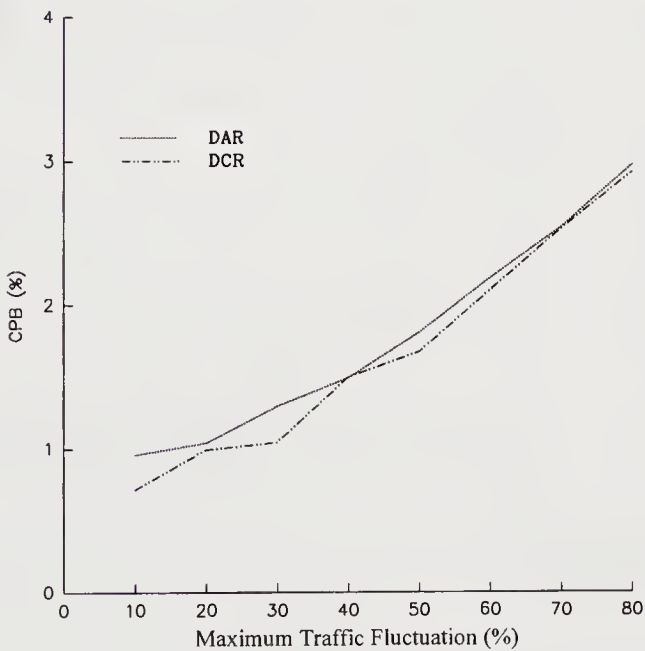


Figure 8b Average CBP versus maximum traffic fluctuation for the DAR and DCR controls.

5.2 Dynamic Traffic Conditions

Under dynamic traffic conditions, we examine the response time of the VPB and VCR controls. The response time of the controls is examined for the theoretical case of a step function, applied to one ATM-SW pair (in one traffic-flow direction). That is, the traffic offered to one ATM-SW pair increases as a step function by 100% in both service-classes.

First, a medium-term VPB control scheme is applied (Logothetis, Shioda, 1995), with a control interval of 30 min. That is, the VPB rearrangement procedure starts every 30 minutes. We assume that the traffic fluctuation occurs at the end of the second control interval (i.e. after 60 min). Bandwidth reservation of 23 bandwidth units is applied to the first service-class. Second, the DAR control is applied which is a decentralized control scheme governing each call arrival. Third, the DCR control is applied with a zero control interval (DCR 0) and, fourth, the DCR control is applied again with 10 sec control interval (DCR 10). Trunk reservation of 48 bandwidth units is applied to benefit the first choice path for all VCR controls. The CBP of each ATM-SW pair is measured every 15 min.

Figure 9 shows the worst CBP of the network versus time. The response time of the VPB control is 75 (135-60) min. VCR controls respond faster than VPB control to absorb the traffic variation because of their very short control interval. The VPB control needs a considerably larger control interval because of the required time for bandwidth rearrangement due to the existing call-connections at the time point of bandwidth rearrangement.

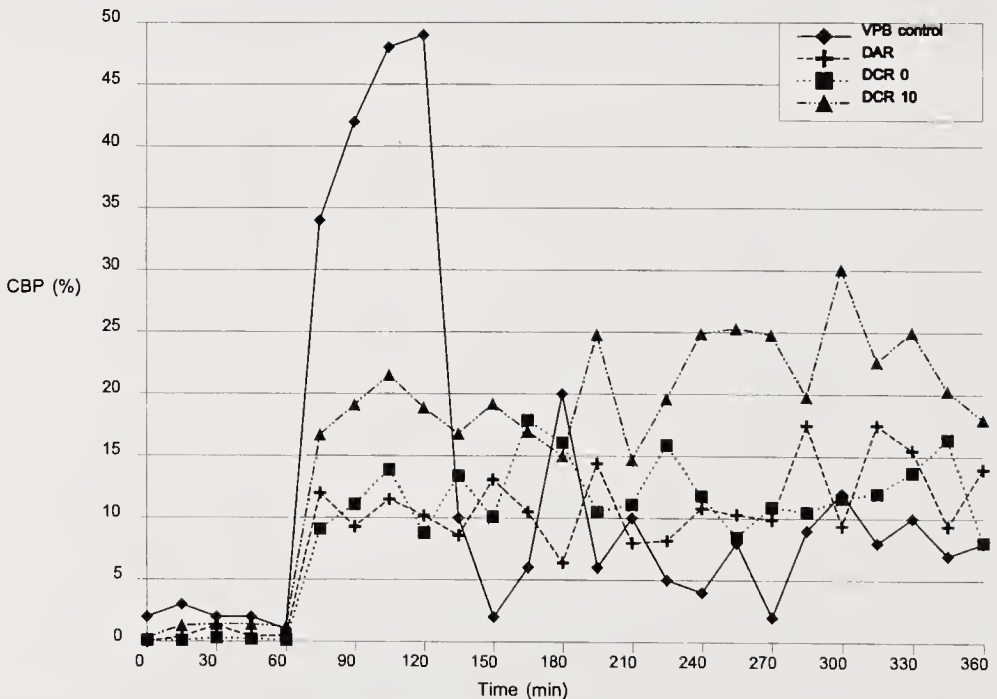


Figure 9 Response time for the VPB and VCR controls.

6 CONCLUSION

Two traffic controls, the VPB control and the VCR control, are presented for ATM networks and the following comparisons are examined:

a) The performance of two VCR control schemes, the DAR and DCR, is examined, considering several trunk reservation parameters. A larger trunk reservation number is needed when the traffic variation among the ATM-SW pairs of the network increases.

b) The same comparison is carried out, considering various control intervals for the DCR control (centralized control). A very small control interval is needed for the DCR control to achieve a better performance than the DAR.

c) Under static traffic conditions, the VPB control is compared with the VCR (DAR and DCR) controls in respect to their effectiveness in minimizing the worst CBP of the network. The worst CBP of the network without any control is shown. The VPB control is more effective than VCR control when the traffic fluctuation is large.

d) Under dynamic traffic conditions, the response time of each traffic control scheme is measured by means of simulation. The VCR control has a faster response time than the VPB control. This is due to the considerably larger control interval required for VPB control. Nevertheless, the response time of VPB control is satisfactory if we consider the network resiliency within two control intervals.

7 REFERENCES

- Ash G.R. (1990), "Design and Control of Networks with Dynamic Non-hierarchical Routing", IEEE Comm. Mag., 34-40.
- Ash G.R. (1985), "Use of a Trunk Status Map for Real Time DNHR", in Proc. on the ITC 11, 795-801.
- Cameron W.H, Regnier J., Galloy P., Savoie A.M. (1983), "Dynamic Routing for Intercity Telephone Networks", ITC-10, 3.2.3.1-3.2.3.8.
- Gibbens R.J., Kelly F.P., P.B. Key P.B. (1989), "Dynamic Alternative Routing - Modelling and Behaviour", ITC 12, 1019-1025.
- Girard A., "Routing and Dimensioning in Circuit-Switched Networks", Addison - Wesley, 1990.
- Kaufman J.S. (1981), "Blocking in a shared resource environment", IEEE Trans on Commun., COM-29.
- Key P.B., Whitehead M.J.(1989), "Cost-Effective use of networks employing Dynamic Alternative Routing", ITC-12, 987-997.
- Key P.B, Cope G.A. (1990), "Distributed Dynamic Routing Schemes", IEEE Commun. Mag., 54-64.
- Logothetis M., Shioda S. (1992), "Centralized Virtual Path Bandwidth Allocation Scheme for ATM Networks", IEICE Trans. Commun. vol. E75-B, no. 10, 1071-1080.
- Logothetis M., Shioda S., Kokkinakis G. (1993), "Optimal Virtual Path Bandwidth Management Assuring Network Reliability", ICC '93, 30-36.
- Logothetis M. (1995), "Optimal Virtual Path Bandwidth Allocation in ATM Networks ", Tutorial Proc. of "Third IFIP Workshop on Performance Modelling and Evaluation of ATM

- Networks", Ilkley, West Yorkshire, UK.
- Logothetis M., Kokkinakis G. (1995), "A Batch-Type Time-True ATM-Network Simulator", 5th International Conference on Advances in Communication and Control, Rethymno, GREECE.
- Logothetis M., Shioda S. (1995), "Medium-Term Centralized Virtual Path Bandwidth Control Based on Traffic Measurements", IEEE Trans. on Commun., Vol 43, No 10.
- Mase K., Uose H. (1989), Consideration on Advanced Routing Schemes for Telecom. Networks, ITC-12, 973-979.
- Mase K., Yamamoto H. (1990), "Advanced Traffic Control Methods for Network Management", IEEE Commum. Mag., 82-88.
- Mitra D., Seery J.B. (1991), "Comparative Evaluation of Randomized and Dynamic Routing Strategies for Circuit-Switched Networks", IEEE Trans. on Com., vol. 39, no. 1, 103-115.
- Monteiro J.A.S., Gerla M. (1990), "Topological Reconfiguration of ATM networks" Proc. GLOBECOM '90, 207-214.
- Ohta S., Sato K., Tokizawa I. (1988), "A dynamically controllable ATM transport network based on the virtual path concept", Proc. GLOBECOM '88, 1272-1276.
- Rengier J., P. Blondeau P. (1983), Cameron H.W., "Grade of Service of a Dynamic Call-Routing System", ITC-10, 3.2.6.1-3.2.6.9.
- Roberts J.W. (1982), "Teletraffic models for the Telecom 1 Integrated Services Network", Proc. of 10th ITC.
- Sato K., Ohta S., Tokizawa I. (1990), "Broadband ATM Network Architecture Based on Virtual Paths", IEEE Trans. on Commun., vol. COM-38, 1212-1222.
- Shioda S., Uose H. (1991), "Virtual Path Bandwidth Control - Method for ATM Networks: Successive Modification Method", IEICE Trans. vol E74, 4061-4068.
- Shioda S. (1994), "Evaluating the Performance of Virtual Path Bandwidth Control in ATM Networks", IEICE Trans. Commun. vol. E77-B, no. 10, 1175-1187.
- Saito H., Kawashima K., Sato K. (1991), "Traffic Control Technologies in ATM Networks", IEICE Trans., vol E74, no 4, 761-771.
- Stacey R.R., Songhurst D.J. (1987), "Dynamic Alternative Routing in the British Telecom Trunk Network", Proc. Int'l Switching Symp. Session B 12.4.1, Phoenix, Arizona.
- Yokoi H., Shioda S., Saito H. and Matsuda J. (1995), "Performance Evaluation of Routing Schemes in B-ISDN ", IEICE Trans. Commun., Vol. E78-B, No. 4.

8 BIOGRAPHY

Ioannis Z. Papanikos was born in Agrinio, Greece, in 1966. He received the diploma in Electrical Engineering from the University of Patras, Patras / Greece, in 1991. He is working towards Ph.D. degree in the Wire Communications Laboratory of the Electrical & Computer Engineering Department of the University of Patras. He has participated in many national research projects of the Wire Communications Laboratory, in the area of telecommunications. His research interests include network management, routing control, and multimedia communications. He is a member of the Technical Chamber of Greece.

Michael D. Logothetis was born in Stenies, Andros, Greece, in 1959. He received the Dipl.-Eng. and Ph.D. degrees in electrical engineering, both from the University of Patras, Patras/Greece, in 1981 and 1990 respectively. From 1982 to 1990, he was a Teaching and Research Assistant at the laboratory of Wire Communications, University of Patras, and participated in many national research programs and two EEC projects (ESPRIT), dealing with telecommunication networks, as well as with natural language processing. From 1991 to 1992, he was Research Associate in NTT's Telecommunication Networks Laboratories. Since 1992, he is a Lecturer in the Department of Electrical Engineering, University of Patras, Greece. His research interests include traffic control, network management, simulation and performance optimization of telecommunication networks. He is a member of the IEEE (Commun. Society - CNOM), IEICE and the Technical Chamber of Greece.

George K. Kokkinakis was born in Chios, Greece, in 1937. He received the Diploma in Electrical Engineering (Dipl.-Ing.) in 1961, the Doctor's Degree in Engineering (Dr.-Ing) in 1966 and the Diploma in Engineering Economics (Dipl. Wirt.-Ing), all from the Technical University of Munich, Germany. During 1968-1969 he served at the Ministry of Coordination in Athens. Since 1969 he is with the Department of Electrical Engineering at the University of Patras, where he has organized and is directing the Wire Communications Laboratory (WCL). His current activity in research and development, which coincides with the activity of WCL, includes the design and optimization of telecommunication networks, and the analysis, synthesis, recognition and linguistic processing of the Greek language. He has published several books and over 100 technical papers, articles and reports on Telecommunications, Electrotechnology and Speech Technology. Prof. Kokkinakis is a senior member of IEEE and a member of the Technical Chamber of Greece (TEE), the VDE (Verein Deutscher Elektrotechniker), the ESCA (European Speech Communication Association), the EURASIP (European Association for Signal Processing), the SEFI (Societe Europeenne pour la Formation des Ingenieurs), the EEEE (Greek Operations Research Society), and the LSA (Linguistics Society of America).

PART FOUR

Adaptation Layer and Protocols

Some simulation results about TCP connections in ATM networks

M.Ajmone Marsan, A.Bianco, R.Lo Cigno, M.Munafò

Dipartimento di Elettronica, Politecnico di Torino

Corso Duca degli Abruzzi 24, 10129 Torino – Italy

Tel. +39 11 5644000 – Fax. +39 11 5644099

email: {ajmone,bianco,locigno,munafò}@polito.it

Abstract

We discuss simulation results concerning the performance of the TCP protocol when running over high-speed ATM networks. Two network topologies are considered: a simple network topology, comprising just two ATM switches and supporting 3 TCP connections, and a candidate Italian ATM network topology comprising ten ATM switches and supporting 6 TCP connections. In all simulation scenarios the TCP traffic is mixed with some background traffic whose level is taken as a variable parameter. Both the background traffic and the TCP traffic are either unshaped, or shaped according to the GCRA algorithm.

The effect of the background traffic on the TCP protocol performance is discussed, varying the buffering capacity within nodes as well as the peak bit rate that each TCP connection is allowed to use. Numerical results clearly show that shaping the TCP traffic according to fixed parameters significantly improves both the goodput and the efficiency of the TCP connections with respect to the case in which no traffic shaping is implemented. Moreover, the performances achievable with an *adaptive* shaping of the TCP traffic (using a simplified version of the ABR ATM transfer capability) can be observed to be extremely satisfactory.

Keywords

ATM, simulation, TCP, traffic control, traffic shaping, ABR

1 INTRODUCTION

The evolution of the ATM standards and products towards the LAN market clearly indicates that the first ATM networks will be mainly used to transport data traffic for business applications. Even in the long run, however, data traffic is expected to remain a relevant part of the load in ATM networks. It is thus very important that the high-level protocols used for the implementation of data applications be carefully investigated with respect to their adaptability to the ATM environment.

TCP (Transport Control Protocol) is today the de facto standard transport protocol for data applications in the LAN, MAN and WAN areas. Many experts believe that TCP

for a long time to come will remain the most frequently used transport protocol in the ATM environment, even if it has been recognized that TCP is not specifically tailored to high bandwidth-delay product networks.

Some studies of the behaviour and performance of TCP when used in ATM networks already appeared in the literature (Romanow 1994, Meempat 1994, Bianco 1994, Ajmone 1995², Perloff 1995). Our work concentrates on the effect that the heterogeneous traffic present in the network, that we call background traffic, may have on the TCP performance. The importance of the presence of background traffic goes beyond the reduction of the bandwidth available to TCP, since background traffic interferes with the TCP behavior by altering the probability of cell losses within node buffers. Moreover, we also investigate the influence of "traffic shaping" on the TCP performance. Shaping the TCP traffic at the network ingress may be a reasonable approach to allow the network to control the TCP source rate, without requiring a substantial rewriting of the TCP code itself. Note however that a negotiation phase between the user and the network is necessary in order to agree on a given peak cell transmission rate; this rate will limit the throughput of the TCP connection, even during periods of low network load, when the throughput achievable by the TCP connection could be higher. A possible solution to this drawback is the use of shaping devices that can adapt the peak cell transmission rate of a TCP source according to feedback signals conveyed by the network. Such a solution was foreseen by the ATM Forum within the ABR (Available Bit Rate) ATM transfer capability (ATM Forum 1995). We investigate the viability of this solution by studying the effectiveness of a simplified version of ABR.

2 PERFORMANCE RESULTS

The results presented in this paper are obtained via simulation with CLASS, an ATM network simulator recently developed at Politecnico di Torino (Ajmone 1995¹). To obtain a model for the TCP protocol, we adapted the officially distributed C code of the BSD 4.3-reno release (Jacobson 1990), without considering the delayed and selective ACK options (for details see Ajmone 1995²). The simulation software was validated by comparison with measurements performed on an experimental ATM LAN; furthermore, an approximate analytical model is being developed for a simple network configuration.

In all the simulation scenarios that we considered, TCP connections are supposed to perform a long file transfer from a TCP transmitter to a TCP receiver: the TCP transmitter sends only data segments, and the TCP receiver returns only ACK segments. TCP sources operate in sustained overload: segments are always ready at the transmitter when an ACK is received. The size of the buffers at the TCP transmitters is set to a large value that avoids any loss at the source during the fragmentation process of a TCP segment into ATM cells. The TCP receivers are assumed to be fast enough and to have enough buffer space so as to avoid losses. The maximum window size is set to a value that allows a single TCP transmitter to obtain the full available bandwidth on the link. It is supposed that the TCP protocol always transmits segments of 9140 bytes (9180 bytes including IP and TCP overhead), the suggested maximum segment size for TCP over ATM; TCP segments are divided in cells by the AAL5 sublayer (requiring the addition of 8 overhead bytes).

The background traffic messages are generated according to a Poisson process, with a

truncated geometric message length distribution with mean equal to 20 cells and maximum length 200 cells; the background traffic is segmented according to the AAL3/4 sublayer.

The burstiness of both the TCP connections and the background traffic can be controlled with a shaping device that operates according to an adaptation of the GCRA (Generic Cell Rate Algorithm) recommended by ITU-T for traffic policing in ATM networks (ITU-T 1992).

A GCRA shaper is based on the control of the cell interdeparture time by delaying cells that are scheduled for transmission too early. The basic parameters of the GCRA shaper are the *bandwidth allocation factor* β , which is the amount of bandwidth allocated to the connection relative to its mean bandwidth, and the *cell delay variation tolerance* τ which is the amount of time that a cell is allowed to “accelerate” with respect to its expected arrival time. When the background traffic is shaped, we assign to each connection $\beta = 1.2$ and $\tau = 0$ in the case of the simple 2-node network, while in the Italian network the bandwidth allocation factor is $\beta = 1.5$.

Numerical results are presented as curves referring to two performance indices:

- the useful throughput, called *goodput*, at the TCP receivers, obtained considering the received data, but discarding all the faulty and the retransmitted segments;
- the *efficiency* of the TCP connections, i.e., the ratio between the goodput and the total offered load of TCP connections.

Curves are plotted as functions of the background traffic load, expressed in Mbit/s of user data; the background load on the link can be computed multiplying the abscissa values by 53/44. The TCP goodput is instead expressed in Mbit/s of user data for uniformity with what is generally done in literature, considering the whole 9180 byte segments. Thus, in order to obtain the link utilization, the TCP goodput must be divided by the efficiency and multiplied by a factor 53/48 (AAL5 is used) and added to the background load. The background traffic is formed by 9 different connections (in addition, an identical background traffic flows on the backward link).

Simulations were either run until the receiver throughput reached a 98% precision with 95% confidence, or stopped after about one minute of simulated time. However, with the exception of the case when ABR-like services are simulated, the 2-node network with 100 Mbit/s background traffic was so overloaded that one of the TCP connections was forced to close by the backoff mechanism, a symptom that the network is not working properly. For this reason the results of the simulation runs with a background traffic load equal to 100 Mbit/s must be interpreted very carefully.

2.1 The two-node network

We first consider a very simple network, whose topology is sketched in Fig. 1, and comprises only two ATM switches. The data rate on each channel is 150 Mbit/s, and channel L_0 , linking the two ATM switches, is the system bottleneck. Three TCP connections share the network resources with a variable amount of background traffic.

The performance of this very simple ATM network was studied in detail in (Ajmone 1995²) as a function of three variables: the TCP connection length (the network span), the background traffic load and characteristics, and the TCP traffic shaping parameters.

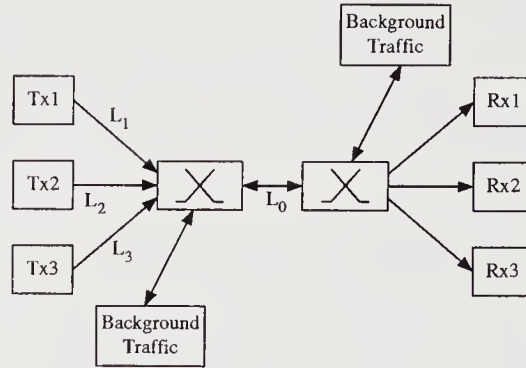


Figure 1 The simulated two-node ATM network

We only report here some figures that help the reader understand the novel results and also serve as a reference when considering more complex scenarios.

The results presented in Fig. 2, are obtained when all the links L_0, L_1, L_2, L_3 in Fig. 1 have length 500 km, while the links from the second ATM switch to the TCP receivers are assumed to have negligible length. The TCP connections length is thus 1000 km. The results are presented as a function of the background traffic load, for two values of the node buffer size in front of the congested link L_0 : 1000 and 5000 cells, with shaped background traffic. As expected, when no shaping is performed on the TCP traffic (dotted lines with square markers), the TCP goodput steadily decreases with increasing background traffic; on the contrary, an increase in the node buffer size results in an increase of the TCP goodput, even if this increase is not very significant, as can be observed comparing the lines with the square markers in the left-hand side figures.

The results when the traffic on TCP connections is shaped are presented on the same charts with the plus and diamond markers for the cases of 50 and 15 Mbit/s shaping, respectively, assuming a cell delay variation tolerance $\tau = 0$. These shaping values correspond to 1/3 and 1/10 of the bottleneck link capacity. The results are rather interesting, showing that smoothing the burstiness of the traffic offered to the network allows TCP connections to better exploit the available resources. In particular, when a 50 Mbit/s shaping is enforced on TCP connections and no background traffic is present, the TCP connections completely saturate the link capacity, since they grab 149.4 out of 150 available Mbit/s, while without shaping the goodput does not exceed 83 Mbit/s, for 5000-cell node buffers. In any case, the goodput achieved with a 50 Mbit/s shaping is always greater than the unshaped goodput, regardless of the node buffer size and the background load.

The situation is slightly different when we analyze the curves with 15 Mbit/s shaping. In this case the TCP goodput is limited by the shaping function, not by the window mechanism, and it remains constant until the background load is increased to 75 Mbit/s. In this case, for high background traffic load, the goodput is greater than the one obtained in the cases without shaping and with 50 Mbit/s shaping. It is interesting to notice that in the case of 100 Mbit/s background traffic load, when the network is clearly overloaded, the performance of the TCP connections is basically the same for the three cases that were considered.

More insight can be achieved by looking at the efficiency of the TCP protocol (the

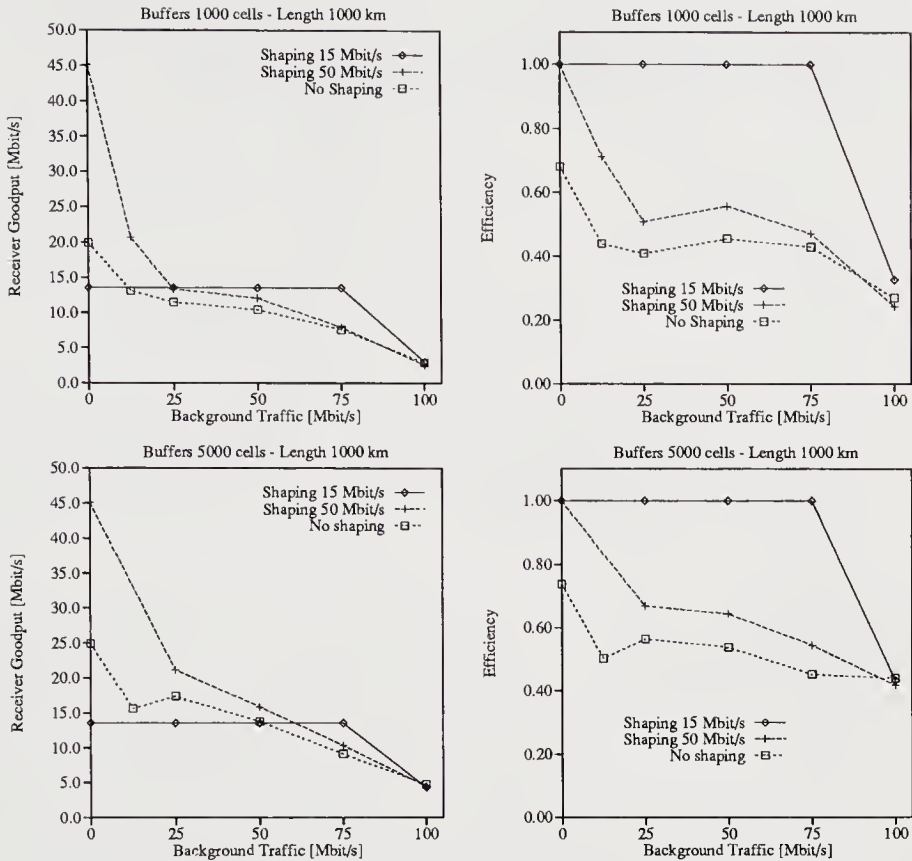


Figure 2 Average goodput and efficiency of the TCP connections for the simulated two-node network with 1000 km connections, as a function of the node buffer size and the background load; the background traffic is shaped

charts on the right-hand side of Fig. 2). These curves clearly show that as soon as the total traffic offered to the network exceeds the bottleneck link capacity, the efficiency of the TCP protocol becomes very poor, dropping to 0.5 or even less. Moreover, the more bursty is the traffic, the poorer is the efficiency. This result is also confirmed by simulation runs without shaping of the background traffic, where the TCP performance (not reported in the graphs) is even poorer. Indeed, the only acceptable, even amazingly good, situation is the one with 15 Mbit/s shaping, whose efficiency remains equal to 1 (no segment loss was recorded) up to a background traffic load equal to 75 Mbit/s; when the background load reaches 100 Mbit/s, the network is, as already stated, overloaded in all cases. It is interesting to notice the fact that with node buffer size equal to 1000 cells, the efficiency of TCP without shaping and with 50 Mbit/s shaping seems to increase slightly for background traffic load 50 and 75 Mbit/s after dropping to about 0.5 for background traffic load 50 Mbit/s. This phenomenon might be due to statistical fluctuations, but we believe that it is more probably due to phasing phenomena like those examined in (Romanov 1994, Bianco 1994).

The second set of results that we discuss considers TCP connections with different

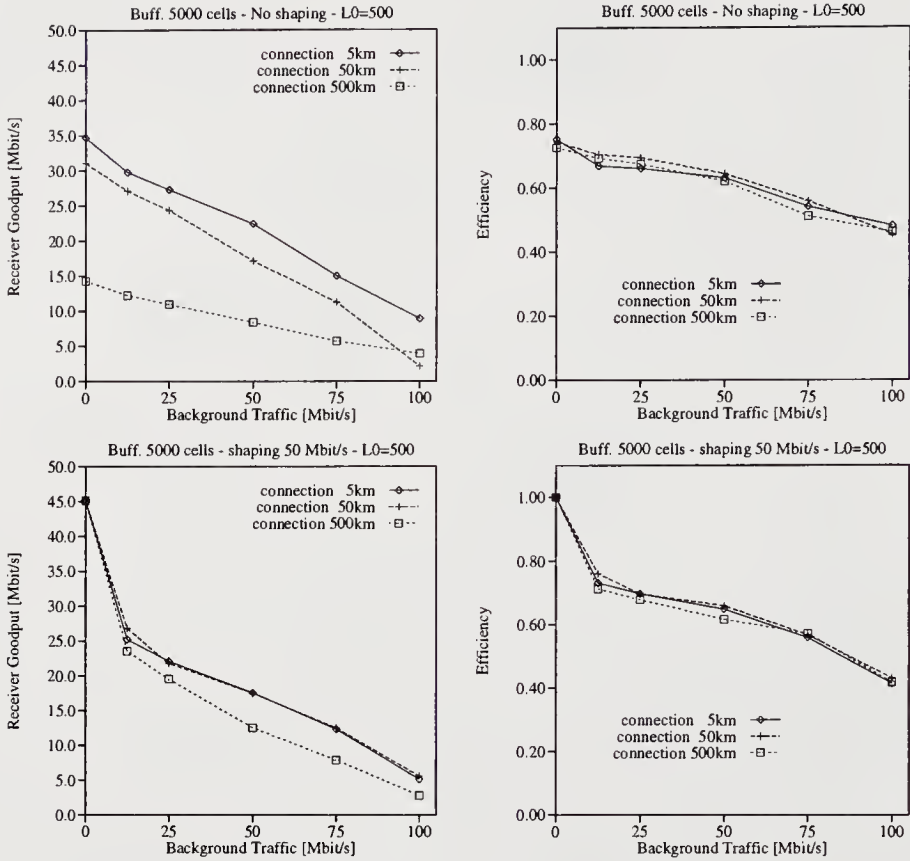


Figure 3 Goodput and efficiency of the TCP connections for the two-node network with 1000, 550, and 505 km connections, as a function of the shaped background traffic load, with buffer size equal to 5000 cells

lengths. This situation may be very common in reality, and it deserves investigation, since the TCP control mechanism is known to be biased against longer connections. In this scenario the goodput of each connection is separately taken into account and plotted. With respect to the simulation scenario presented in Fig. 1, the bottleneck link length L_0 is set to 500 km, while the lengths of the links L_1 , L_2 and L_3 are set respectively to 5, 50 and 500 km, resulting in connections whose lengths are 505, 550 and 1000 km. Fig. 3 reports the results for buffer size equal to 5000 cells, when the TCP connections are either unshaped or shaped at 50 Mbit/s. The shaping at 15 Mbit/s is not reported for the sake of brevity since all of the connections obtain exactly the same goodput.

When the TCP connections are unshaped, it can be noticed that the goodput obtained by the connections is inversely proportional to the connection length, as expected, since the TCP throughput is roughly inversely proportional to the round trip delay. The unfair behavior is clear, and it can be remedied by adopting a 50 Mbit/s shaping. In this case the goodput difference between the connections of length 505 and 550 km is negligible, while the connection with length 1000 km still gets a lower bandwidth, but with low background traffic loads this difference becomes less significant.

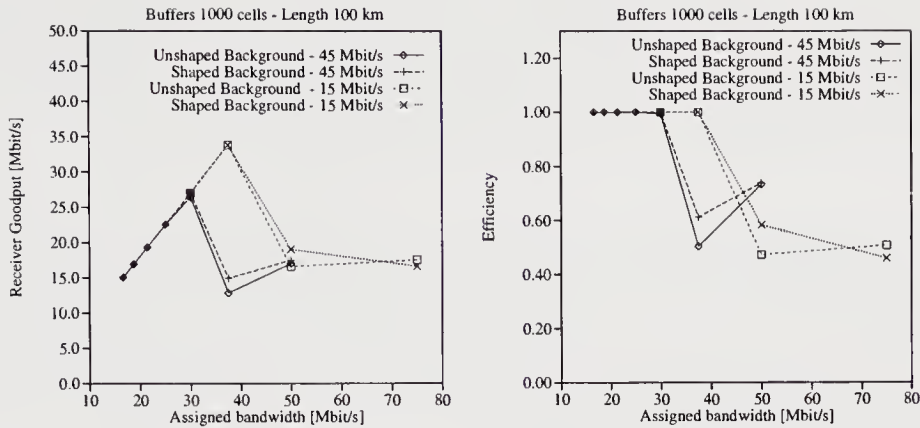


Figure 4 Goodput and efficiency of the TCP connections for the two-node network with three 100 km connections, as a function of the assigned bandwidth, with 1000-cell buffers; the background traffic load is set to either 15 or 45 Mbit/s

It is interesting to notice that the efficiency of the connection is independent from the connection length, even if no shaping is performed on the connections. This means that the losses due to buffer overflow are roughly proportional to the bandwidth grabbed by the connection.

Let us now consider what happens if we draw the results as a function of the bandwidth assigned to each TCP connection. In this case, with reference to Fig. 1, we set all the links lengths to 50 km, thus simulating 100 km connections. The size of the buffer in front of the congested link L_0 is set to 1000 cells. Fig. 4 reports the results obtained in this scenario. Each graph contains two pairs of curves: the first pair is obtained with a background traffic level equal to 45 Mbit/s, while the second one is obtained with background traffic level equal to 15 Mbit/s. Curves within each pair refer either to the case of shaped background traffic or to the case of unshaped background traffic.

The difference between the curves obtained by shaping the background traffic and those obtained by letting the background traffic remain unshaped is negligible because the buffer is big enough to accommodate the background traffic bursts. All curves show the following behaviour: if the assigned bandwidth is such that the link is not overloaded (assigned bandwidth up to 30 Mbit/s with 45 Mbit/s background, and up to 37.5 Mbit/s with 15 Mbit/s background), then the TCP connections efficiency sticks to one and hence the average goodput increases linearly. As soon as the link becomes overloaded, the TCP efficiency drops to about 0.5 and the goodput decreases.

The results presented in Fig. 4 show that, at least statically, it is possible to identify a shaping rate that optimizes the throughput obtained by TCP connections as a function of the background load; this same value also allows the maximum exploitation of the network resources without QoS reduction. This consideration suggests the exploitation of *adaptive* shaping algorithms, like those specified in the ABR ATM transfer capability, (ATM Forum 1995) for the transport of TCP connections.

In order to investigate the performance of the TCP protocol on an ABR-like transfer capability, a simplified version of ABR was implemented in the CLASS simulator. This

implementation follows the ATM Forum guidelines (ATM Forum 1995) but considers only the key aspects of the algorithms, neglecting all the details that are believed not to play a key role in determining the system performance. For this reason, in order not to create confusion, we shall refer to this scenario as *adaptive shaping*, rather than ABR.

The main features of adaptive shaping are as follows.

- The interaction between the end users and the network is managed through special RM (resource management) cells that are transmitted in band. RM cells convey only a ternary feedback to sources: either Increase Rate (IR), or Keep Rate (KR), or Decrease Rate (DR). This ternary feedback is intended to guide the behavior of the source shaper.
- TCP sources shape their traffic, and introduce in their cell flow one RM cell every N_{RM} information cells; the feedback of the RM cell is always initialized to IR by the source.
- TCP receivers route RM cells back toward their corresponding sources, without changing the feedback carried in the RM cells.
- ATM switches monitor the traffic on the forward link, trying to identify any congestion situation. However, switches can modify the feedback in RM cells only when these reach the switch in their backward trip, while returning to the source (this is done in order to reduce the distance and hence the delay between the control point and the source). The feedback in an RM cell can be modified only from IR to either KR or DR, or from KR to DR. This is done in order to avoid the danger that nodes closer to the source set the feedback to more optimistic values than nodes farther away, that may be experiencing congestion.
- ATM switches determine their congestion state by monitoring the occupancy of the buffer associated with the link on which forward RM cells are routed. Congestion is determined using two thresholds T_l and T_h . If the buffer is filled below T_l then the switch does not modify the feedback carried in RM cells, that thus keeps its current value (IR if not previously reduced by other nodes); if the buffer is filled above T_h then the node sets the feedback in RM cells to DR; if the buffer is filled between the two thresholds then the feedback value is set to KR (unless it was already set to DR, in which case it remains DR).
- Source shaping devices always follow the indication that is contained within an RM cell; the time needed to adapt the rate is negligible. The transmission rate R_T can only be set to a value that divides the capacity of the link, i.e., $R_T = \frac{C}{N_S}$ where C is the capacity of the link and N_S is an integer. When a source shaping device receives an RM cell with a DR feedback, the value of N_S is increased by 1; instead, when an RM cell carries an IR feedback, the value of N_S is decreased by 1. This introduces a quantization effect that in some cases, especially when the bandwidth requirements of the connections are high, may affect the performance of the system, introducing oscillations in the transmission speeds.
- Switches are able to enforce fairness in the partition of the bandwidth among connections.

Numerical results were obtained with $N_{RM} = 32$ (one RM cell every 32 information cells), and by setting the two thresholds T_l and T_h to 1%, and 50%, respectively, of the switch buffer size.

Fig. 5 reports the results obtained with adaptive shaping in the two-node network sce-

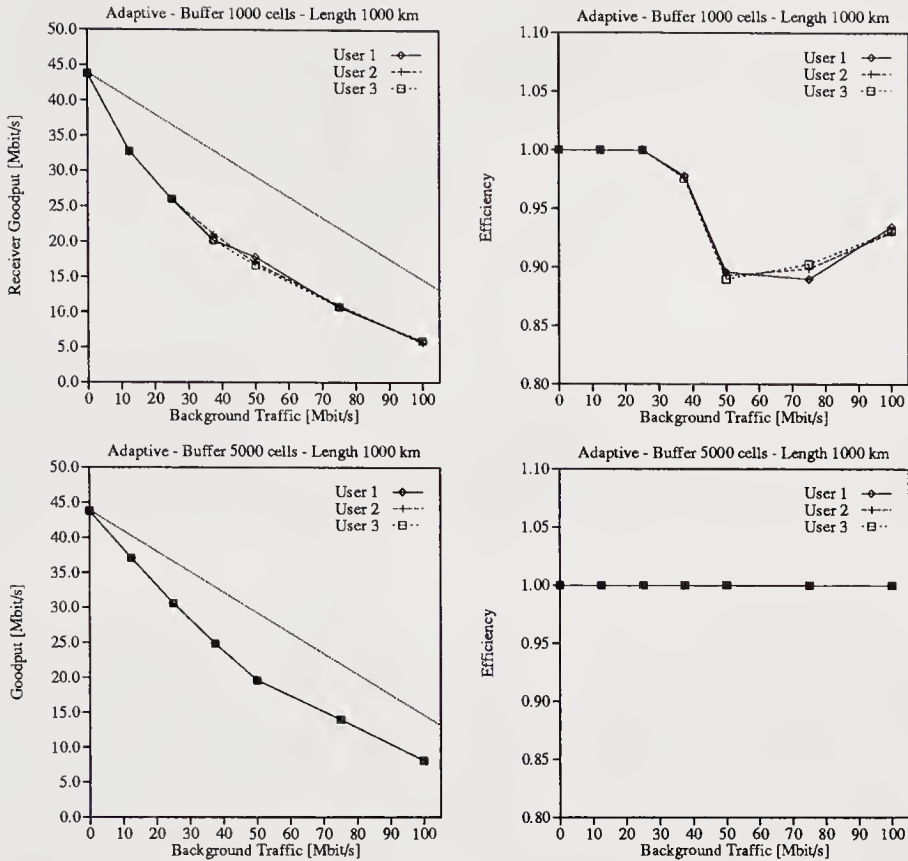


Figure 5 Goodput and efficiency of the TCP connections for the two-node network with adaptive shaping, with 1000 km connections, as a function of the background load, with buffer size equal either to 1000 or to 5000 cells

nario with 1000 km connections. These results are to be compared with those reported in Fig. 2; however, here throughput and efficiency are plotted separately for each TCP connection. The dotted straight line represents the available free bandwidth for each connection (obtained by subtracting the background and RM traffic loads from the available data rate, dividing the result by the number of TCP connections, and multiplying by 48/53 to account for the ATM cell overhead). The maximum allowed transmission rate is 50 Mbit/s for each TCP connection. Observe that the scale of the efficiency plots is greatly magnified with respect to the one in Fig. 2. The performance improvements that can be obtained with adaptive shaping are quite remarkable, and the great increase in efficiency must be noted in particular. A further important consideration is that the performance is now much better with 5000 cell buffers than with 1000 cell buffers: the reason for this difference is that these buffers must be large enough to absorb the transient phase between the congestion detection and the transmitter adaptation, whose duration is proportional to the network span. The efficiency in the case of 1000 cell buffers is affected by the already mentioned granularity in the shaper rates. Indeed, at medium loads the transmission rate keeps oscillating between roughly 15 Mbit/s and 50 Mbit/s; a trans-

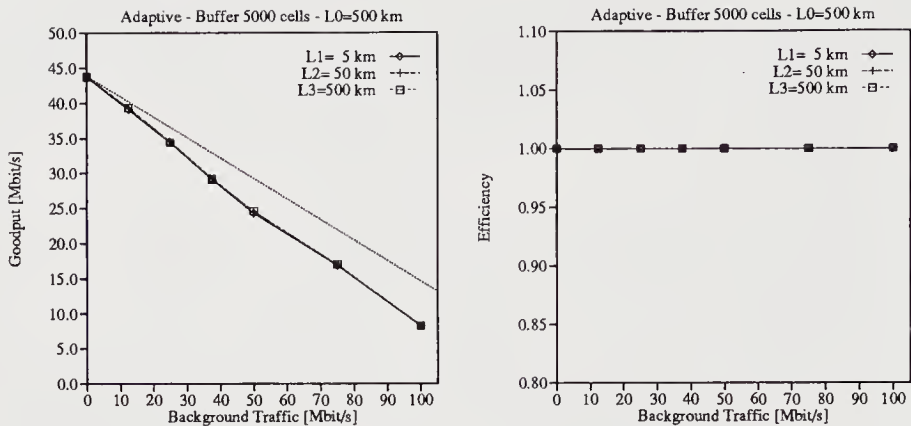


Figure 6 Goodput and efficiency of the TCP connections for the two-node network with adaptive shaping, with 1000, 550, and 505 km connections, as a function of the shaped background traffic load, with buffer size equal to 5000 cells

mission rate of 50 Mbit/s is clearly too much for the network load, but since the allowed rates above 25 Mbit/s are only 30, 37.5 and 50 Mbit/s, the transmission rate is increased too fast, thus leading to buffer overflows. This phenomenon is attenuated at high loads because the transmission rates are lower and the rate granularity becomes negligible.

Fig. 6 reports the results obtained with adaptive shaping when the TCP connections have different lengths. This figure can be compared with Fig. 3. In this case the advantage of a mechanism that allows the transmission rate to be controlled, and the fairness among connections to be enforced, leads to a striking performance improvement: all TCP connections have efficiency one, all of them receive the same amount of bandwidth which corresponds to a very high fraction of the available bandwidth. It is quite interesting to notice that the overall performance in this case is better than the one obtainable when all TCP connections are 1000 km long. Since shorter connections are easier to control, this means that all connections benefit from the presence of short connections: a behavior which is exactly the opposite of the one observed in the case of TCP connections without adaptive shaping.

2.2 The Italian network

The candidate Italian network topology comprises ten ATM switches, located in the major Italian cities, and is shown in Fig. 7; the buffering capacity at all nodes is set either to 100 or to 1000 cells per port, and the user transmission buffer sizes are set to quite large values in order to avoid losses at the source. Six TCP connections can be identified: Mi-Ro, To-Fi, Ve-To, Ro-Ba, Ba-Pa and Pa-Ba. The total amount of background traffic in the network is equal to 0.8, 1.0 and 1.2 Gbit/s, and the traffic distribution is highly asymmetric, the network having essentially two hot spots in Rome and Milan. The complete workload distribution for the background traffic is reported in Table 1. Table 2 presents the background load, measured as a percentage of the link capacity, of all the links crossed by the six TCP connections.

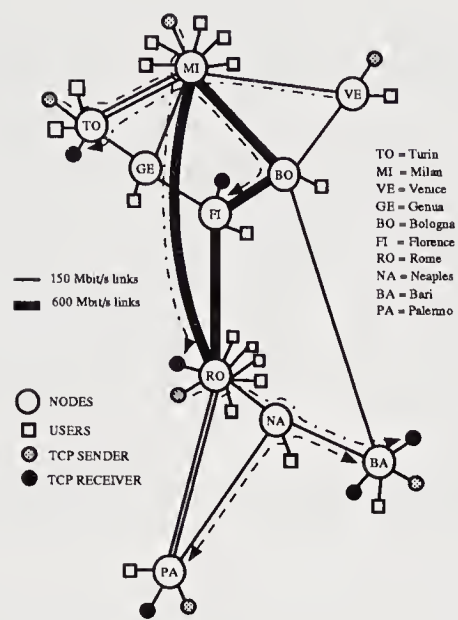


Figure 7 Topology of the Italian network

Node	MI	TO	GE	VE	BO	FI	RO	NA	BA	PA
MI	0	52	20	2	18	3	206	17	2	17
TO	52	0	3	0	3	5	32	3	0	3
GE	20	3	0	0	2	2	12	1	0	1
VE	2	0	0	0	0	0	1	0	0	0
BO	18	3	2	0	0	2	11	1	0	1
FI	35	5	2	0	2	0	22	2	0	2
RO	206	32	12	1	11	22	0	10	1	10
NA	17	3	1	0	1	2	10	0	0	1
BA	2	0	0	0	0	0	1	0	0	0
PA	17	3	1	0	1	2	10	1	0	0
Total	369	101	41	3	38	70	305	35	3	35

Table 1 Traffic matrix used in the simulation of the Italian topology; the traffic is generated by the node in the column and goes to the node in the row; the relations are expressed in thousandths of the global generated traffic

TCP Conn	0.8 Gbit/s			1.0 Gbit/s			1.2 Gbit/s		
	1st	2nd	3rd	1st	2nd	3rd	1st	2nd	3rd
Mi-Ro	0.41			0.51			0.61		
To-Fi	0.294	0.114	0.11	0.367	0.143	0.13	0.44	0.165	0.15
Ve-To	0.072	0.30		0.090	0.377		0.098	0.43	
Ro-Ba	0.25	0.06		0.32	0.074		0.38	0.088	
Ba-Pa	0.06	0.028		0.074	0.035		0.088	0.042	
Pa-Ba	0.028	0.06		0.035	0.074		0.042	0.088	

Table 2 Measured load of the background traffic, as a percentage of the link capacity, on the links crossed by the TCP connections

All the TCP connections have a window size that allows a transmission speed of up to 150 Mbit/s if no shaping is applied by the source. The Mi-Ro TCP connection is carried on a link with much available capacity, since either 60%, or 50%, or 40% of a 600 Mbit/s channel is available for it, respectively, in the three cases of background load. The Ba-Pa and Pa-Ba connections are running on very lightly loaded links, while the three TCP connections To-Fi, Ve-To, and Ro-Ba run over 150 Mbit/s channels whose loads could significantly influence the TCP performances. The Ro-Ba and Pa-Ba connections interfere with one another on the Na-Ba link.

We present results for either shaped or unshaped background traffic. Simulations were run considering five possible scenarios for all the TCP connections: unshaped TCP connections, TCP connections shaped at a link speed equal to either 25Mbit/s, or 37.5Mbit/s, or 50Mbit/s, which means that the TCP goodput can be at most either 22.5, or 33.8, or 45.1 Mbit/s, and finally TCP connections with adaptive shaping. All these scenarios are simulated considering node buffer lengths of either 100 or 1000 cells.

Figures 8 and 9 present the goodput and efficiency for the six TCP connections, as a function of the global background network load; curves refer to TCP connections shaped either at 25 Mbit/s, or at 37.5 Mbit/s, or at 50 Mbit/s, or unshaped, with background traffic either shaped or unshaped; the node buffers are 100 cells long. Figures 10 and 11 reports the results for the same scenarios, but with node buffer size 1000 cells.

The first consideration concerns the buffering capacity within ATM switches. When the buffers are quite small (100 cells) the burstiness of the background traffic has a great impact on the performance of the TCP connections. Instead, when the buffering capacity is large enough to absorb the bursts of cells generated by the background traffic sources (1000 cells buffers), the influence on the TCP connections of the burstiness of the sources becomes negligible.

Let's now consider each one of the six connections separately, since all of them exhibit peculiar behaviours that are worth discussing.

The Mi-Ro connection has a lot of spare bandwidth to exploit; thus it operates with

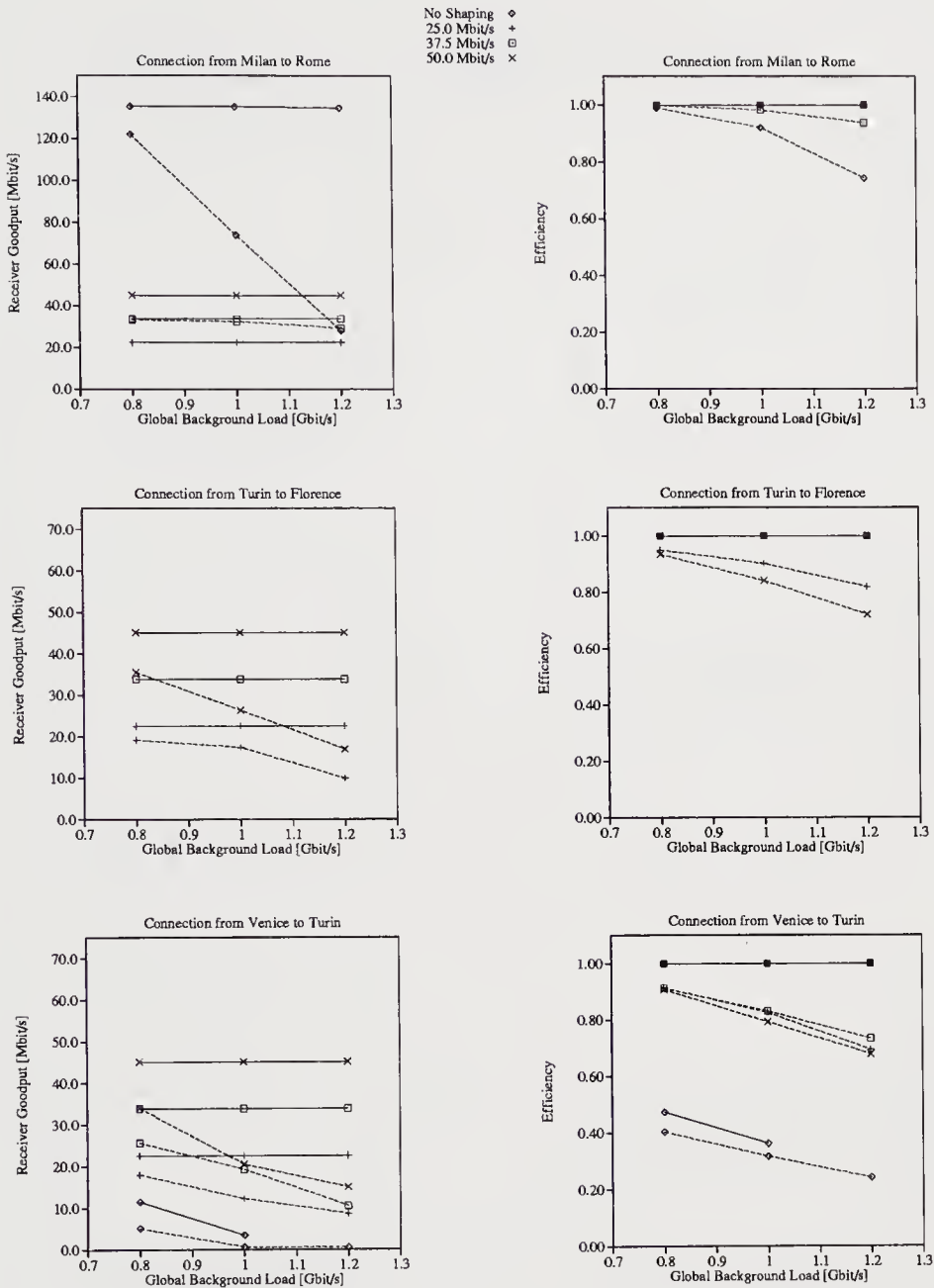


Figure 8 Average goodput and efficiency of the TCP connections for the Italian network as a function of the background load, when the node buffers sizes are 100 cells; solid lines refer to shaped background traffic, whereas dashed lines refer to unshaped background traffic

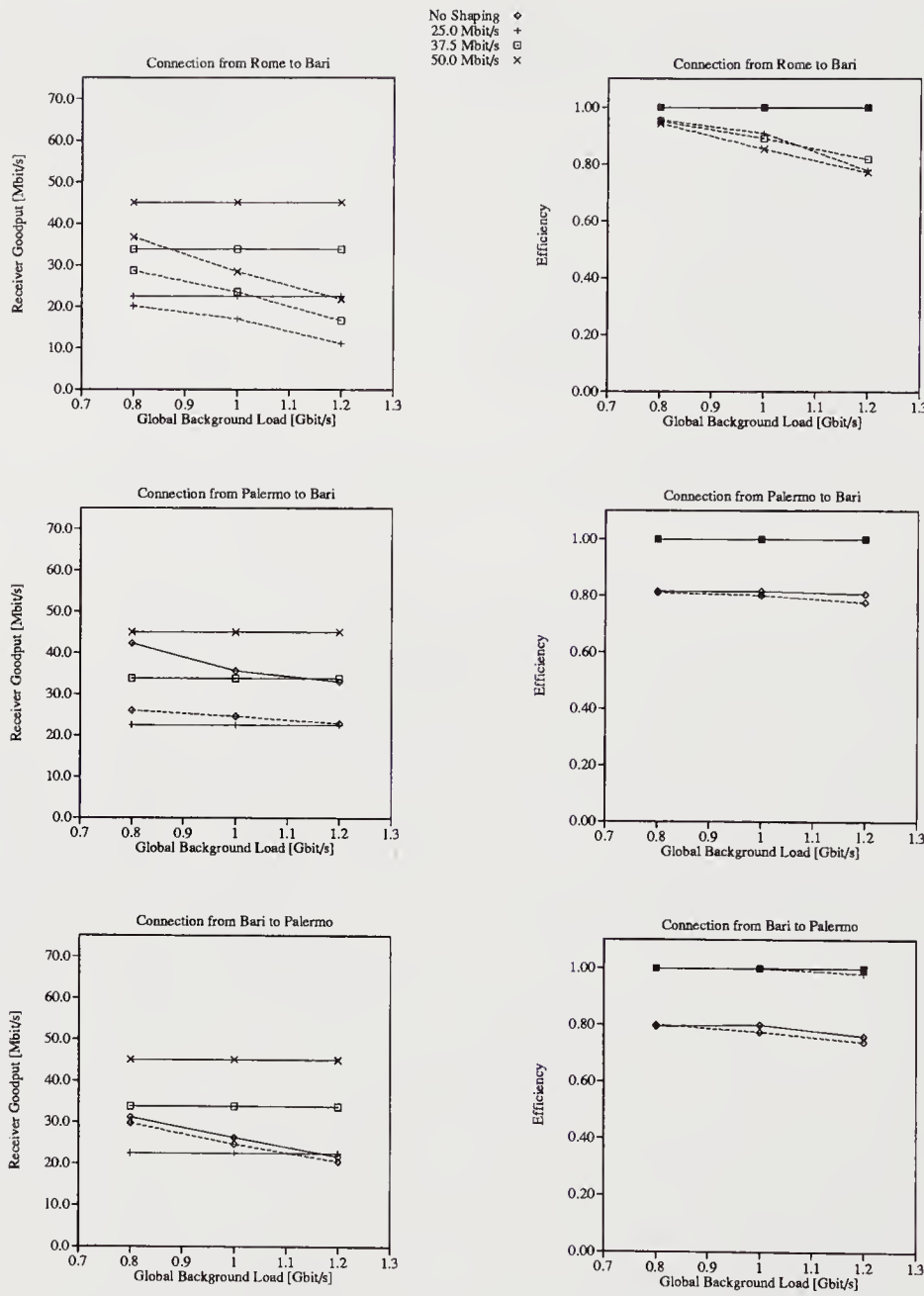


Figure 9 Average goodput and efficiency of the TCP connections for the Italian network as a function of the background load, when the node buffers sizes are 100 cells; solid lines refer to shaped background traffic, whereas dashed lines refer to unshaped background traffic

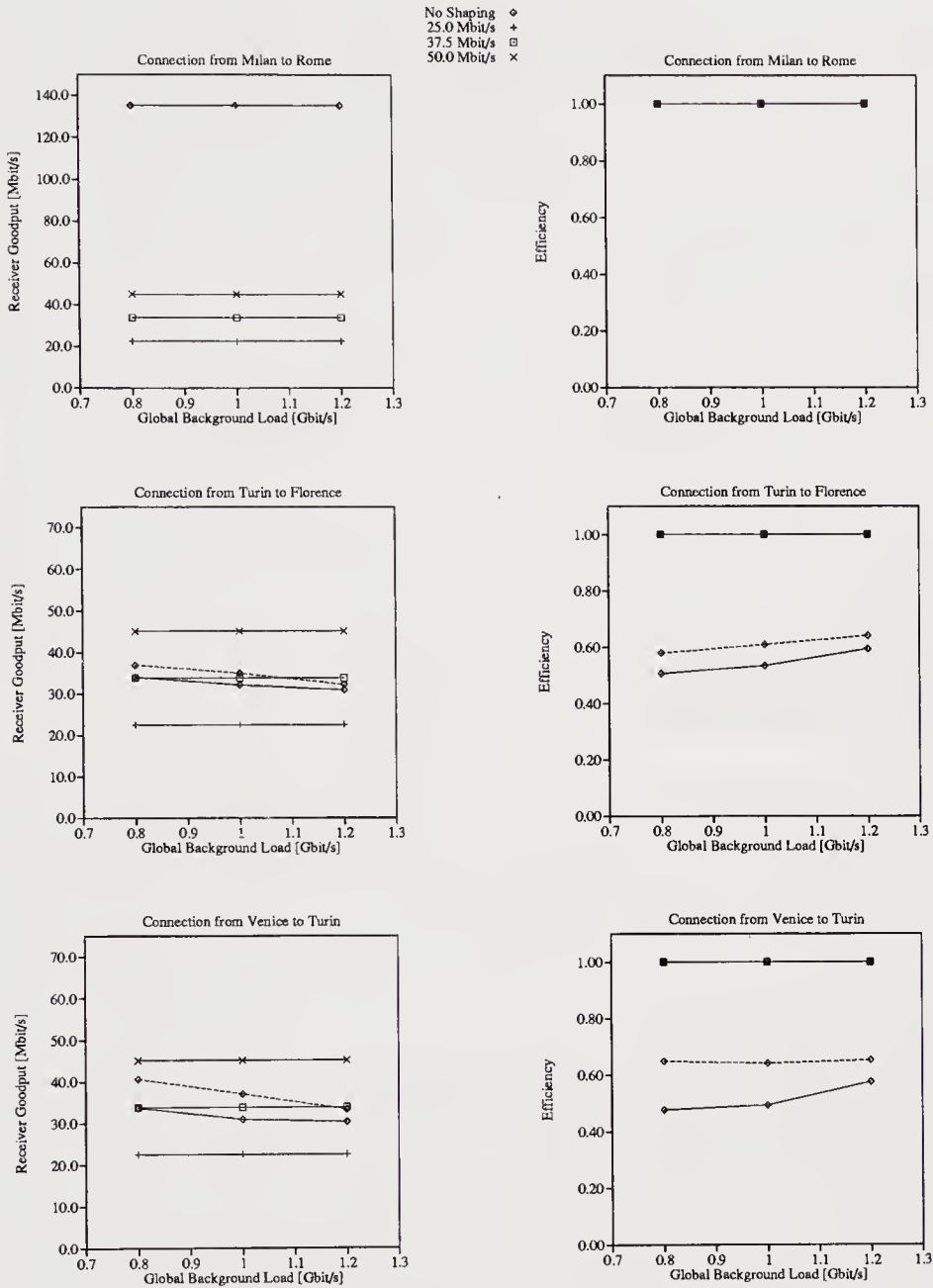


Figure 10 Average goodput and efficiency of the TCP connections for the Italian network as a function of the background load, when the node buffers sizes are 1000 cells; solid lines refer to shaped background traffic, whereas dashed lines refer to unshaped background traffic

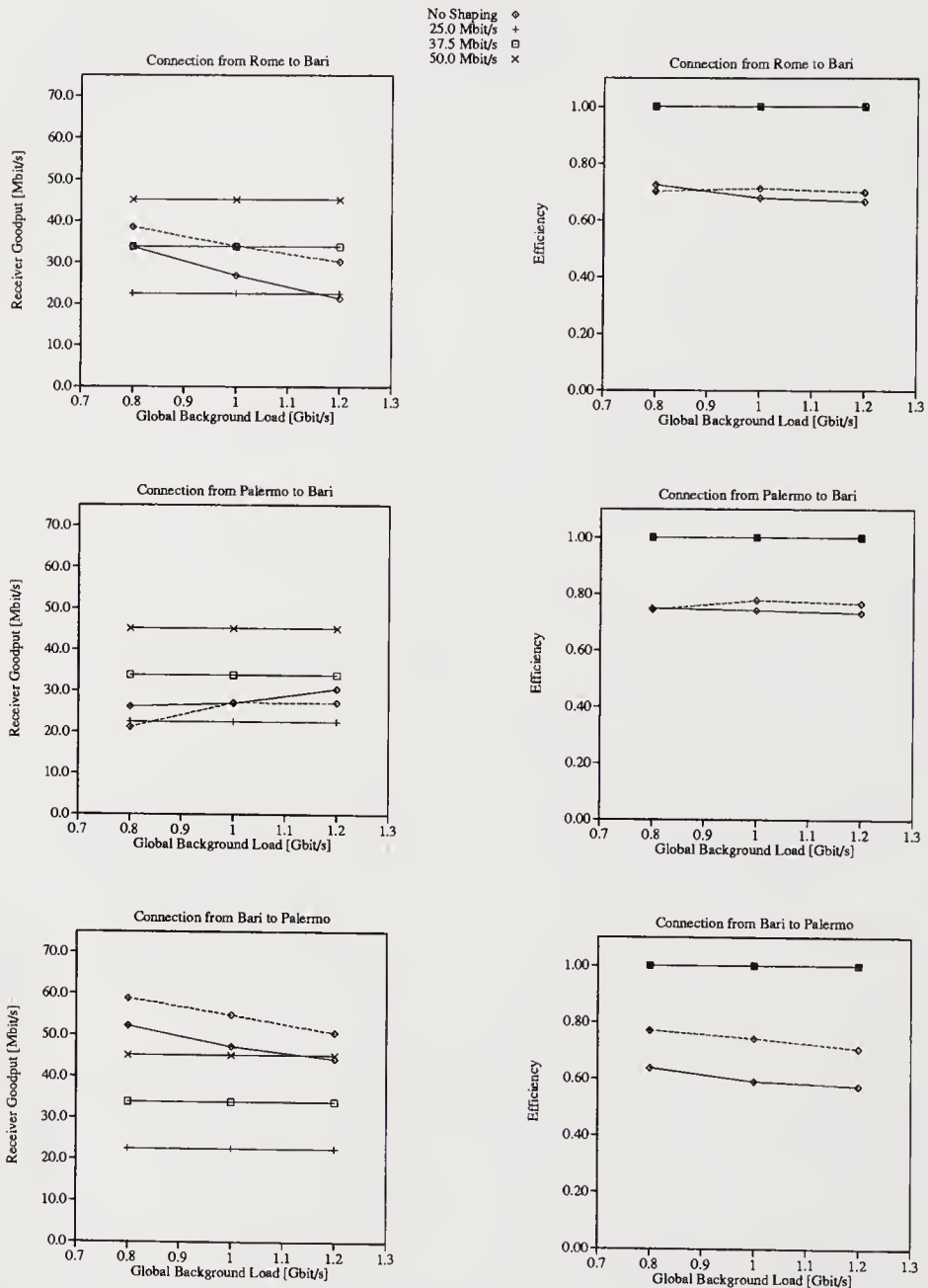


Figure 11 Average goodput and efficiency of the TCP connections for the Italian network as a function of the node buffer size and the background load, when the node buffers sizes are 1000 cells; solid lines refer to shaped background traffic, whereas dashed lines refer to unshaped background traffic

efficiency one, grabbing all the resources it can, in all case except one: if neither the TCP traffic nor the background traffic are shaped and the node buffers are small (100 cells), losses occur in the node buffer, so that both the TCP goodput and efficiency significantly decrease with the increase in the background load.

The To-Fi and Ve-To connections have a similar amount of available resources to exploit, and they behave similarly. Both of them completely use their assigned bandwidth when both their traffic and the background traffic are shaped, or when the node buffers are large enough to absorb the background traffic burstiness. On the other hand, both connections suffer significantly when their traffic is not shaped, obtaining very poor efficiency and a remarkable reduction in goodput. It is important to observe that the missing points in the graphs, like for instance those referring to the To-Fi connection without shaping of the background and TCP traffics with 100 cells buffers, correspond to simulations where the TCP connections were closed due to the TCP backoff mechanism. This behavior clearly shows that without some kind of rate control the TCP flow control mechanism is not able to work properly in high speed networks. The last effect to be observed is that when the buffers size is 1000 cells and the TCP traffic is not shaped, TCP achieves better performance if also the background traffic is not shaped; this is due to the fact that when the background traffic is not shaped, cells are lost in bursts, thus concentrating the losses on a smaller number of TCP segments.

Let's now come to the connections that interact in Naples: Ro-Ba and Pa-Ba. Also in this case some points are missing due to the closure of TCP connections, and the same general considerations presented above apply here too. Moreover, it can be observed that the interaction between the two TCP connections on a lightly loaded link does not seem to jeopardize performance.

Finally, consider the Ba-Pa connection. This connection runs alone on a lightly loaded set of links, and its performance is consequently quite good. The most interesting aspect to be noted in this case is what happens to the TCP connections when no shaping is used, and the buffer size is increased from 100 to 1000 cells. The goodput of the connection increases significantly with the larger buffers, but the efficiency remains very poor, below 0.8, and in fact it is even reduced when the buffer size increases. This phenomenon once again confirms that TCP by itself is not suited to high speed networks, since it wastes a great amount of resources.

Figure 12 refers to the case of adaptive shaping, with the characteristics described in the previous section, but with no enforcement of fairness among connections. The circular markers refer to the case of 100 cell buffers, while the square markers refer to the case of 1000 cell buffers; black markers refer to unshaped background traffic, white markers to shaped background traffic. The buffer thresholds are set to $T_l = 5$ and $T_h = 50$ in the case of buffer size 100, and to $T_l = 10$ and $T_h = 500$ in the case of buffer size 1000. The maximum transmission rate is 75 Mbit/s for all connections, except Mi-Ro that is allowed to transmit up to 150 Mbit/s. First of all, it must be noted that when buffers are 1000 cells long all connections attain efficiency 1, a result which is quite a success in itself; moreover, the throughput of the connections is higher than that obtained without shaping or with fixed shaping. Also in the case with 100 cell buffers the benefits of an adaptive shaping policy are quite evident: both the efficiency and the throughput are higher than without shaping. However, in this case buffer sizes are not large enough to allow a smooth control of the sources, i.e., they cannot accommodate all the cells that are transmitted at high speed by the source before it receives the RM cell with the DR feedback, hence the

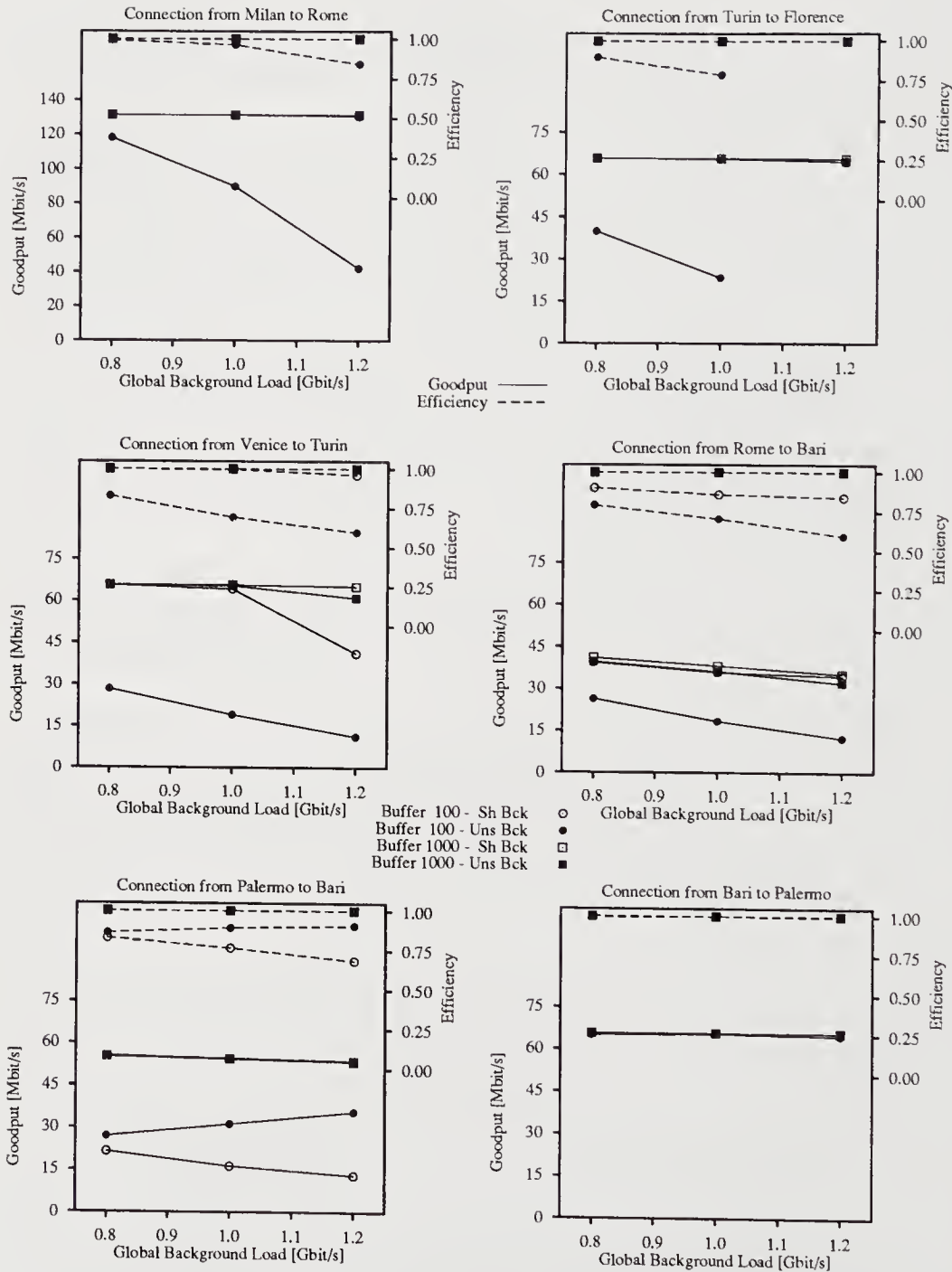


Figure 12 Average goodput and efficiency of the TCP connections in the case of adaptive shaping for the Italian network as a function of the node buffer size and the background load

efficiency of the TCP connections is not always one. Notice that in these conditions the To-Fi connection is still forced to close when the background load is 1.2 Gbit/s. One more aspect worth noticing is the behavior of the Pa-Ba connection with buffer size 100 cells, without shaping of the background traffic. In this case the TCP connection throughput increases when the background increases. This is due to the fact that for this connection the true bottleneck is the Na-Ba link, where most of the traffic is due to TCP, while the competing TCP connection also suffers from the quite heavily loaded Ro-Na link, hence when the background traffic increases the Ro-Ba connection is forced to reduce its rate and the Pa-Ba connection can exploit the bandwidth on the Na-Ba link freed by the Ro-Ba connection. This shows that an adaptive shaping is indeed able to exploit dynamically the available bandwidth. Moreover, it seems that, if the number of connections competing for the bandwidth is small, an adaptive shaping scheme may work well even if fairness is not enforced by the network.

3 CONCLUSIONS

The performance of the TCP protocol when running over ATM networks was studied through simulation, in two network scenarios, considering the TCP connections goodput and efficiency as significant performance parameters.

The variable parameters of the study are the background traffic load, the buffering capacity in the ATM switches, the traffic shaping parameters of both TCP and background traffic and the length of the TCP connections.

Numerical results clearly show that shaping the traffic on the TCP connections greatly improves the TCP performance, and also indicate that ABR-like solutions may be quite advantageous.

An important advantage of the shaping approach for TCP connections is that shaping techniques can be applied with no modification of the TCP protocol itself.

4 REFERENCES

- Ajmone Marsan M., Bianco A., Do T.V., Jereb L., Lo Cigno R., Munafò M. (1995¹) ATM Simulation with CLASS. *Performance Evaluation Journal*, **24**, 137-159.
- Ajmone Marsan M., Bianco A., Lo Cigno R., Munafò M. (1995²) Shaping TCP Traffic in ATM Networks. *IEEE ICT'95*, Bali, Indonesia.
- ATM Forum/95-0013R2 (1995), ATM Forum Traffic Management Specification 4.0.
- Bianco A. (1994) Performance of the TCP Protocol over ATM Networks. *IEEE ICCCN'94*, San Francisco, CA USA.
- ITU-T Recommendation E.371 (1992) Traffic Control and Congestion Control in B-ISDN. Geneva, Switzerland.
- Jacobson V. (1990) Berkeley TCP evolution from 4.3-tahoe to 4.3-reno. *Eighteenth IETF*, Vancouver, BC Canada.
- Meempat G. (1994) Interactions of Reliable Stream Data Transport Protocols with Rate Control Algorithms. *IEEE ICCCN'94*, San Francisco, CA USA.
- Perloff M., Reiss K. (1995) Improvements to TCP Performance in High-Speed ATM Networks. *Communications of the ACM*, **38-2**, 90-100.

Romanow A. and Floyd S. (1994) Dynamics of TCP Traffic over ATM Networks. *ACM SIGCOMM'94*, London, UK.

5 ACKNOWLEDGMENTS

This work was supported in part by a research contract between Politecnico di Torino and CSELT, in part by the EC through the Copernicus project 1463 ATMIN, and in part by the Italian Ministry for University and Research.

6 BIOGRAPHIES

Marco Ajmone Marsan is a Full Professor at the Electronics Department of Politecnico di Torino, in Italy. He holds a Dr. Ing. degree in Electronic Engineering from Politecnico di Torino, and a Master of Science from the University of California, Los Angeles. Since November 1975 to October 1987 he was at the Electronics Department of Politecnico di Torino, first as a Researcher, then as an Associate Professor. Since November 1987 to October 1990 he was a Full Professor at the Computer Science Department of the University of Milan, in Italy. His current interests are in the fields of performance evaluation of data communication and computer systems, communication networks and queueing theory.

Andrea Bianco is a Research Assistant at the Electronics Department of Politecnico di Torino. He holds a Dr. Ing. degree in Electronics Engineering and a Ph.D. in Telecommunications Engineering both from Politecnico di Torino. Since May 1991 to December 1994 he has been with the Electronics Department of Politecnico di Torino as a Ph.D. student. In 1993 he spent one year visiting HP Labs in Palo Alto, CA, working on performance analysis of the TCP protocol over ATM networks. Since November 1995 he is with the Dipartimento di Elettronica of the Politecnico di Torino. His current research interests are in the field of access protocols for all-optical networks, performance analysis of ATM networks, and formal description techniques.

Renato Lo Cigno is a Research Engineer in the Electronics Department of Politecnico di Torino. He received a Dr. Ing. degree in Electronics Engineering from Politecnico di Torino in 1988. Since then he has been with the telecommunication research group of the Electronics Department of Politecnico di Torino, first under various research grants and contracts, then as a staff member. His research interests are in communication networks simulation and performance analysis.

Maurizio Munafò is a Research Engineer in the Electronics Department of Politecnico di Torino. He holds a Dr. Ing. degree in Electronics Engineering and a Ph.D. in Telecommunications Engineering, both from Politecnico di Torino. Since November 1991 he has been with the Electronics Department of Politecnico di Torino, where he has been involved in the development of an ATM networks simulator. His research interests are in simulation and performance analysis of communication systems.

PART FIVE

Network Management

Feedback & Pricing in ATM Networks

L. Murphy

*Department of Computer Science and Engineering,
Auburn University, AL 36849, USA.*

tel: +1 334 844-6326; fax: +1 334 844-6329

email: lmurphy@eng.auburn.edu

J. Murphy

*School of Electronic Engineering,
Dublin City University, Glasnevin, Dublin 9, Ireland.*

tel: +353 1 704-5444; fax: +353 1 704 5508

email: murphyj@eeng.dcu.ie

Abstract

Admission control and congestion control can provide traffic guarantees in ATM networks. However some users may not be able to describe their traffic accurately enough for the network to provide such guarantees. By sending a dynamic feedback signal about the current utilisation of network resources, the network could provide loss guarantees to users who respond appropriately, even without prior traffic descriptors. One possible feedback signal is a price per unit of network resource, based on the network load level : when the load is high, the price is high, and when the load is low, the price is low or zero. We outline a distributed iterative pricing algorithm, and show through simulations that it can simultaneously increase both network and economic efficiency. We also explore some arguments often raised against usage-sensitive pricing, and provide some counter-arguments.

Keywords

ATM Networks, Pricing, Dynamic Feedback, Congestion Control

1 INTRODUCTION

Asynchronous Transfer Mode (ATM) has been adopted as the transfer mode for the Broadband Integrated Services Digital Network (BISDN), e.g. de Prycker (1993), a service-independent network capable of supporting all the communication services that users now require or may require in the future. ATM is also emerging as a local area net-

working technology, since it provides flexible bandwidth-on-demand and internetworking capabilities for conventional data communications. ATM networks are therefore expected to accommodate a wide range of users, including some whose applications require **guarantees** on cell loss and/or delay. These guarantees could be deterministic worst-case or less stringent statistical guarantees. Some users may be satisfied with best-effort service, for which the network offers no guarantees on loss or delay.

Admission control and congestion control can provide performance guarantees and are therefore two of the most important ATM network functions. In order to obtain these guarantees from the network, users have to describe their traffic inputs by specifying values for network-defined **traffic descriptors** such as peak cell rate (PCR) or sustainable cell rate (SCR). However some users may not be able to describe their traffic accurately : because their applications cannot be sufficiently well-characterised by the given traffic descriptors, or because their actual traffic inputs depend on factors outside user control (such as the number of active applications competing for access to a server). A common assumption in many proposed admission control schemes is that traffic which is not well-described cannot get specific guarantees beyond the level of service being provided to best-effort traffic.

The ATM Forum has recognised the problem of providing guarantees to users whose traffic cannot be well-described, and in response has developed a specification for Available Bit Rate (ABR) service, e.g. Ramakrishnan (1995). Users who choose ABR service receive feedback from the network about the current level of network resource utilisation, and can get cell loss guarantees¹ if they respond appropriately – by reducing their input rates in times of congestion, for example.

ABR service is therefore suitable for users whose applications are flexible with respect to delay but not necessarily to loss. This flexible behaviour represents a tool that network operators can use to increase network utilisation while continuing to serve guaranteed traffic such as CBR and VBR applications. In addition, this type of network feedback could modify an adaptive user's traffic at the source rather than after it has been injected into the network. This would help to localise the effects of feedback to the edges of the network and allow simpler internal network operation.

Most suggestions for supporting ABR service assume that well-described traffic which requires performance guarantees gets priority in the use of network resources such as bandwidth or buffer space, and that the remaining resources are fairly shared among the ABR users. Two issues which are not explicitly addressed are

- why more “demanding” traffic should get priority over ABR traffic;
- what constitutes “fair” sharing. Should the available bandwidth be shared equally among all ABR users, for instance ? Or should it be shared according to the various application requirements ?

¹No specific delay guarantees can be provided, so ABR users must be prepared to absorb delays at the traffic source before being allowed to input traffic into the network.

It is important to note that, just because such issues are not addressed explicitly, does not mean that these suggestions are neutral on what are often regarded as *policy issues*. On the contrary : sharing the available bandwidth equally among all ABR users values all such traffic equally, although the users themselves may put widely differing values on their service; giving CBR and VBR users priority over ABR users ignores the possibility that ABR users may value network access more than users with well-described traffic sources. We are not saying that these assumptions are wrong or undesirable, but instead we advocate allowing the users themselves to resolve these issues.

Admission control and congestion control in ATM are difficult problems which so far have not been satisfactorily solved. Two key questions are

- **how should congestion be defined and measured ?** This is a difficult question because individual user requirements vary considerably, so that one user may think the network is congested while another does not; and because in internetworks the responsibility for detecting congestion may be distributed among several network operators, each of which applies a different test at their bottleneck points.
- **how should limited resources be allocated under congestion ?** Some proposals call for users to indicate the relative priority of their traffic – leading to the problem of providing incentives so that all users will not choose the highest priority.

Our aim in this article is to propose a dynamic feedback control scheme which explicitly addresses these issues.

2 DIFFERENT TYPES OF EFFICIENCY

A network is as good, or as bad, as its users perceive it to be. This leads to the conclusion that network performance should be measured in terms of overall user satisfaction with the service they receive. Network engineering measures (such as average packet delay or loss rate) are inadequate reflections of user satisfaction when user requirements vary widely.

Due to the difficulty in accounting for individual user's requirements, however, aggregate network-oriented performance measures are usually used in design and operations problems. Usage is divided into classes according to application requirements and traffic characteristics; for example, real-time video, real-time audio, or off-line file transfer. Each class is regarded as having a single representative user for analytical and control purposes, and class objectives are used to drive the network control process. Therefore the loop is not closed all the way to the users when making operational decisions.

We propose to bring the users *back into the loop* and thereby ensure that performance measures are user-oriented, as shown in Figure 1. A user-oriented network control scheme would take user valuations into account : the network could serve higher-value users even under congestion by temporarily denying access to lower-value users. Such time-

smoothing would not upset users who can tolerate longer delays, while it would improve the network's value to users who get greater benefits from immediate access.

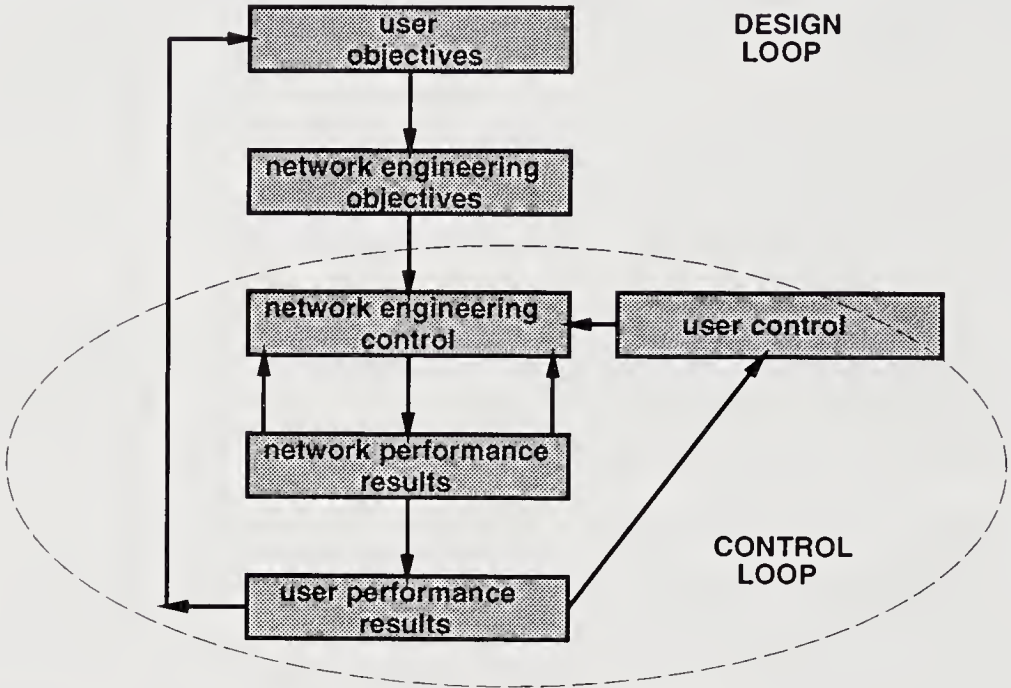


Figure 1 Network design and control loops.

Each user privately decides how much they value network access; our scheme involves giving them incentives to do this. Users would gain by obtaining service more closely matched to their needs; network operators would gain through improved network utilisation and increased user satisfaction with the service they receive. We hope to achieve the same (or better) network performance as with conventional congestion control and resource allocation schemes, while at the same time increase the total value of the network from the users' point of view. Network engineering measures will continue to be important, but we believe that user preferences should be the primary consideration driving resource allocation and congestion control schemes.

We need to distinguish two very different notions of efficiency :

- **Network efficiency** refers to the utilisation of network resources such as bandwidth and buffer space.
- **Economic efficiency** refers to the relative valuations the users attach to their network service.

If a network can maintain an acceptable level of service while minimising the resources necessary to provide this service, we say that its operation is network efficient. If no user

currently receiving a particular Quality of Service (QOS) values it less than another user who is being denied that QOS, we say that operation is economically efficient.

An obvious question is, why will either type of efficiency continue to be important ? Some observers have suggested that the widespread deployment of fibre optic lines, and continuing exponential decreases in processor and memory costs, will result in these network resources becoming essentially “free” so that efficiency in their use will not be important in the future, and all users can always be accommodated. We do not believe these arguments apply in the short or medium terms, if indeed they will ever apply. User demands are increasing exponentially, so that it is not clear when – if ever – network resources will be “free”. Experience suggests that application developers will have no difficulty in designing new services that use up all available resources, perhaps after an initial adjustment period. And market economics dictates that commercial network operators should be aware of the differing valuations that users attach to the same level of network performance. The same considerations apply to privately owned or operated networks : the ultimate goal will continue to be to maximise some measure of the *value* of using the network.

2.1 IMPROVING EFFICIENCY WITH FEEDBACK

Users with flexible traffic inputs can help to increase network efficiency if they are given appropriate feedback signals. When the network load is high, the feedback should discourage these users from inputting traffic; when the load is low, the feedback should encourage them to send any traffic they have ready to transmit. Instead of regarding their load as fixed, the network uses the flexibility of these users as part of a congestion control and avoidance strategy. One possible feedback signal is a price based on the level of network load : when the load is high, the price is high, and when the load is low, the price is low or zero.

Similarly, by associating a cost measure with network loading, all users can be signalled with the prices necessary to recover the cost of the current network load. Price-sensitive users – those willing and able to respond to dynamic prices – increase economic efficiency by choosing whether or not to input traffic according to their individual willingness to pay the current price. Users who value network service more will choose to transmit, while those who value it less will wait for a lower price.

Price signals thus have the potential to increase both network and economic efficiency, though whether a particular pricing scheme increases either notion of efficiency depends on the implementation. One important point needs to be clarified :

- contradictory though it sounds, a scheme based on pricing principles does **not necessarily involve money**. For example, in a private network where one organisation controls all the users, or in a company’s virtual private network, the “prices” are simply control signals. In this case, the users’ applications could be programmed to obtain a desirable traffic mix, to enforce priorities, or to achieve

some other objective.

We envisage that the charge to a user in an ATM network might have many components, such as a connection fee, a charge per unit time or per unit of bandwidth, premium charges for certain services, and so on. We suggest that there should also be a usage-sensitive component during congestion, to increase both network and economic efficiency. *We propose charging only when network congestion indicates that some users may be experiencing QOS degradation, with the size of the charges related to the degree of congestion.* If the network is lightly loaded and all users are getting acceptable QOS, the usage-sensitive prices would be zero.

We recognise that many people are concerned about the use of pricing in network operations. Concerns range from questions about the feasibility and overhead of usage-sensitive pricing, to policy issues such as profit opportunities and fairness. We believe that a clear understanding of the nature of what is being proposed is necessary on all sides. Therefore we first outline our proposed dynamic pricing scheme and some preliminary simulation results, and then address some of the objections often raised in discussions of dynamic network pricing.

2.2 DISTRIBUTED ITERATIVE PRICING ALGORITHM

It is important to note that our proposed pricing algorithm would only be applied to adaptive users, who are able and willing to respond to dynamic prices during a connection by changing their offered traffic. All other users would be charged according to another pricing scheme. How to co-ordinate the various pricing schemes to achieve some overall objective (such as fairness) is a complex issue and we do not address it in this article.

The network and its users are considered to form an economic system. The system has various resources such as link bandwidths and buffer spaces that can be used to meet user demands for service. Network constraints such as buffer sizes or link capacities are translated into cost functions on the demands for resources. The basic property of these cost functions is that marginal cost should go to infinity as usage of the resource approaches capacity.

Each adaptive user is viewed as placing a benefit, or willingness-to-pay, on the resources they are allocated. Given a price per unit of bandwidth or buffer space, a user's benefit function completely determines that user's traffic input. A benefit function could follow the usual economic assumption of diminishing incremental benefit as more of the resource is consumed, see Figure 2(a). Or the user could apply a threshold rule, or series of threshold rules, for deciding how much of the resource to request based on the current price, see Figure 2(b),(c).

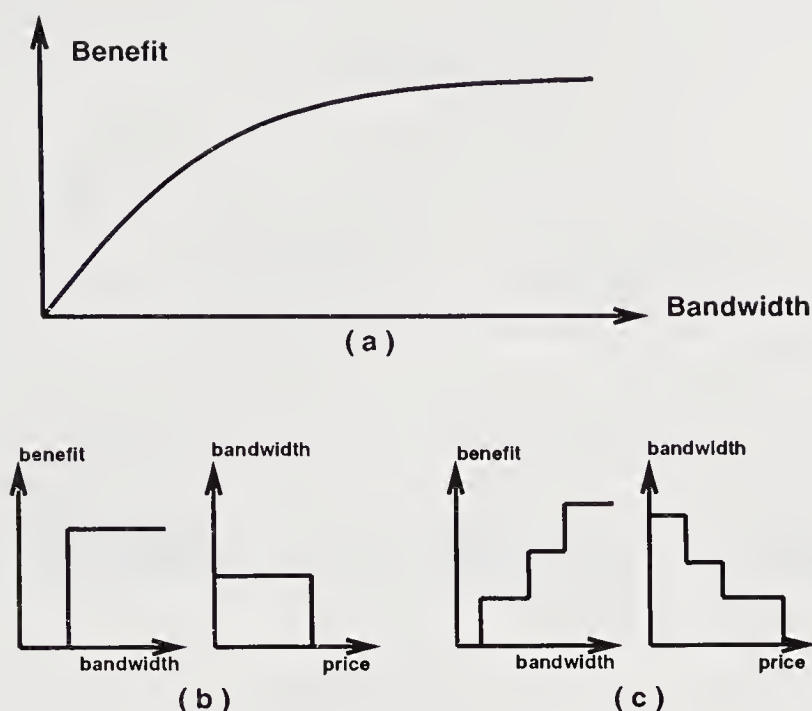


Figure 2 Possible user benefit functions.

The network operator sets the prices so that the marginal benefit the users place on their resource consumption is equal to the marginal cost of handling the resulting traffic in the network². The network operator dynamically adjusts the prices based on current network conditions. It turns out that it is not necessary for the network operator to know the user benefit functions; therefore our pricing scheme is suitable for public as well as private networks.

Time is divided into feedback intervals, within each of which the prices and user benefits are fixed. This model allows users to potentially change their benefit functions every feedback interval, to reflect their satisfaction with the level of service received or their time constraints on having their cells accepted into the network, so the examples in Figure 2 are for a particular interval. Similarly the network re-calculates the prices every feedback interval to reflect current resource usage³.

A distributed iterative pricing algorithm for adaptive users has been developed, e.g. Murphy (1994), Murphy and Posner (1994), see Figure 3. The computation required per iteration at each user and ATM access switch is simple, which suggests that inexpensive processing elements may be sufficient in executing the algorithm.

²These prices only address the variable costs corresponding to network constraints.

³The network and the users may use prediction in their decisions without invalidating this model.

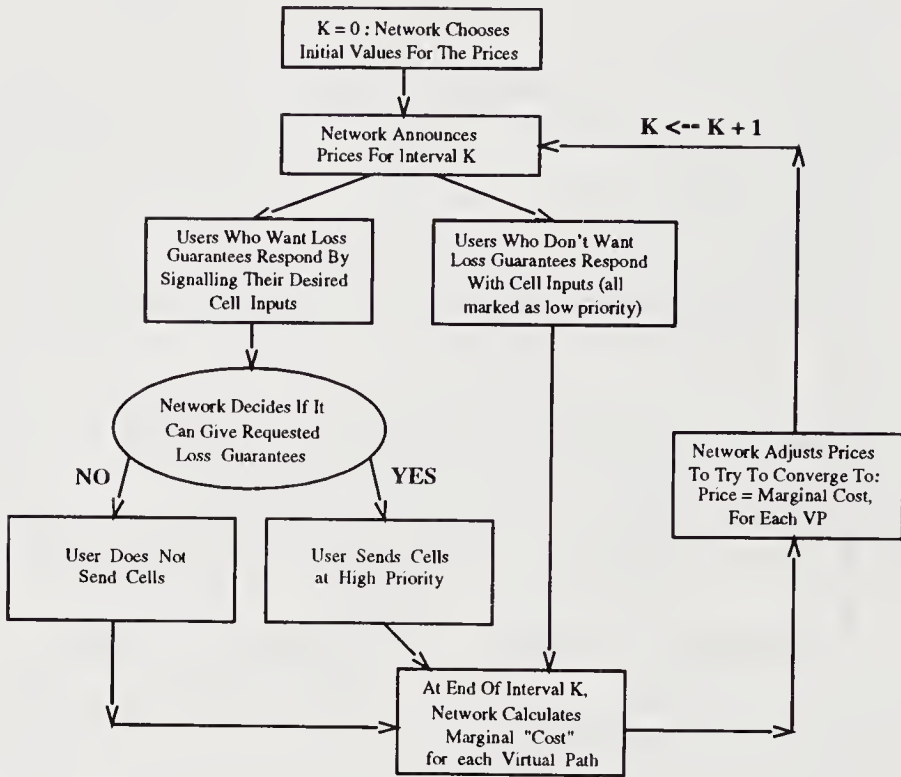


Figure 3 Distributed iterative pricing algorithm for adaptive users.

There are several different types of adaptive users, e.g. Murphy (1995), depending on whether the user is flexible with respect to loss, delay, or both. So far we have modelled two types of adaptive user :

- *Inelastic.* This user requires a delay bound on their traffic, but can tolerate sending only a fraction of the cells that are ready to transmit in the current interval. We assume that cells not sent in the interval are useless to the user and are discarded. For example, this might be the second level of a two-level video codec. The first level contains the minimum necessary information, and would be transmitted as a non-adaptive application. The second level consists of enhancement information. It is not essential that all of the information be delivered; however, a delay guarantee is required — if the enhancement information does not arrive before the playback point, it is considered useless.
- *Elastic.* This type of user waits until feedback from the network indicates that they can input traffic, then transmits and requires that their cells are not lost in the network. Each elastic user decides individually what their transmission criteria are, e.g. the maximum price per cell they are willing to pay. An example of an elastic user type would be a non-real-time data transfer with no ARQ capability, where already-transmitted cells are not buffered at the sender.

The mathematical models used for inelastic and elastic users are described in Murphy (1996), along with the equations governing their responses to the dynamic prices from the network.

3 SIMULATION MODELS & RESULTS

The simulated network is a high-speed ATM 155 Mbps link shared by inelastic and elastic users. Video sources are modelled as inelastic users; data sources are modelled as elastic users. The link model is shown in Figure 4. The network and source models were simulated using SES/*workbench*, e.g. SES (1992), a discrete-event simulator that allows hardware and software simulation. The models were mainly created by use of its graphical user interface. SES/*workbench* compiles the graphical code to C and creates an executable. The simulation execution platform was a cluster of Sparc-10 workstations.

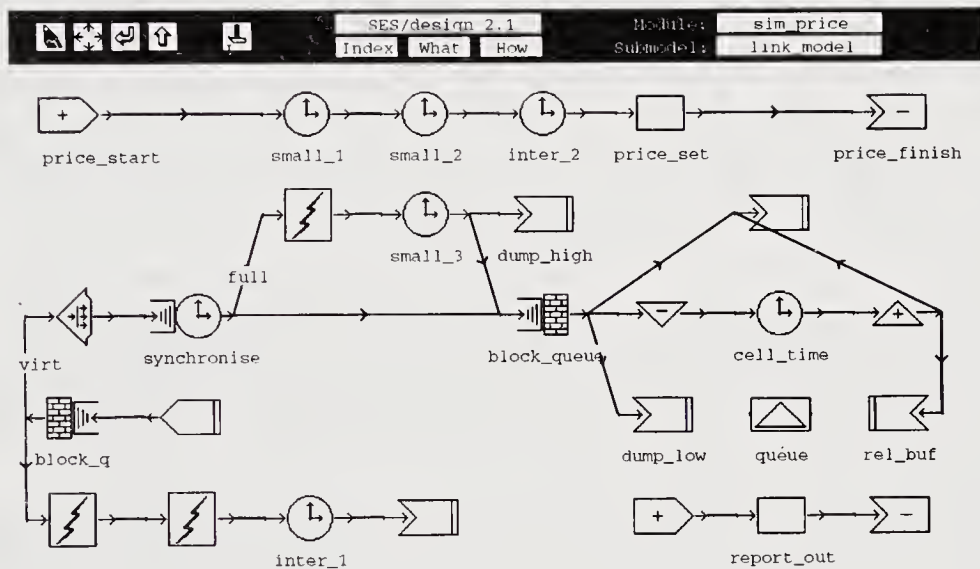


Figure 4 Simulation model for economic efficiency.

The simulation model is made up of submodules, each of which performs a well defined function. The sources generate cells which are input to a network interface submodule. The network interface takes the source bit stream and forms ATM cells. The cell stream from an interface is then input to the ATM switch buffer submodule. This submodule smooths the arrival of cells to the ATM network and so takes care of cell scale congestion. The switch buffer is the limited network resource in our model.

The video source model that we use here is a standard one for video conferencing, e.g. COST (1993). The codec has a compressed bit rate of 2.3 Mbps, which adheres to the

H.261 standard for video, e.g. Murphy and Teahan (1994). There are 20 video sources, each with a mean of 2.3 Mbps and a peak of 5 Mbps. These are input at a rate of 30 frames per second, all synchronised together, so that the inelastic users are (more or less) stationary on the millisecond scale.

The elastic users can be thought of as one user with a lot of files to transfer independently, many users each with one file, or some combination of these types. The negotiations for file transfer or connection set up will only occur when a new video frame is to be sent, i.e. every 1/30 second, because this simplifies the simulation and makes it possible to speed up the run time. Therefore the network renegotiates the PCR every 1/30 second with the elastic users. An empirical distribution for file size ranges was obtained from actual files stored on one of our computers. In the simulations a range was chosen according to this empirical distribution, and then a file size was chosen from a uniform distribution within this range. The peak-to-mean ratio of this source can be high with values up around 1000. The number of data sources in use is taken from a uniform distribution between 1 and 39 sources. Each file to be sent is taken from a uniform distribution between 20 and 660 cells. The average bit rate of a single data source is therefore about 4.3 Mbps.

In our proposed scheme a price is generated by the network based on the present state of the network buffer, and the sources adapt their demands based on this price. What we propose and simulate adheres to the UNI 3.0 specification from the ATM Forum, e.g. ATM Forum (1993). A leaky bucket is also implemented on top of our scheme so that if there is cell loss we can discard the marked ones first.

The model takes in cells over a feedback interval and gives a price to all the sources sharing the link. The price reflects the congestion (if any) in the buffer and hence on the virtual path. The feedback interval is short compared to the video frame time : a value of about 0.05 of a frame time was chosen. To achieve feasible run times we neglect cell scale effects. This neglecting of cell scale effects is critical to the speed up of the simulations. The total utilisation of the link is high, at a value of around 0.85.

Our results are shown in Table 1 and show the difference between using pricing and no pricing. What can be seen is that both network efficiency and economic efficiency increase at the same time by using pricing. Cell loss drops from 19% to under 2%, while the net benefits perceived by the users increase by nearly 15%.

Table 1 Performance and Economic Gains from User Feedback.

Source Type		% Loss	User Value	% Dec. Loss	% Inc. Value
Unpriced	Inelastic	0	240	91.0	14.8
	Elastic	30.4	146		
	Combined	19.1	386		
Priced	Inelastic	4.4	239		
	Elastic	0.1	204		
	Combined	1.7	443		

4 CONCERNS ABOUT USAGE-SENSITIVE PRICING IN NETWORKS

We explore some common arguments against usage-sensitive pricing in network operations in this Section, and provide some counter-arguments. Some previous work along these lines is contained in MacKie-Mason (1994).

- once a network is installed, any load-dependent costs of transferring data are minimal – the fixed costs of network management and maintenance dominate. These fixed costs can be efficiently recovered through connection fees and capacity prices. Why implement an elaborate pricing mechanism to recover the relatively small variable costs ?
 - *Counterpoint* : this ignores the congestion cost which one user's traffic imposes on other users sharing the resources. Bandwidth or buffer space occupied by one user's traffic is not available to other users. When this reduces other users' quality of service (through increased delays, loss rates, blocking probabilities, and so on), they suffer congestion costs which may translate into significant actual costs of service degradation. One mechanism to capture these costs is a price which is sensitive to some indicator of congestion, such as load.
- even if we want to consider congestion costs, how can the network determine what actual costs the current load is imposing on users who probably have widely varying service requirements ? Getting users to reveal these costs is likely to be extremely complicated, if not impossible.
 - *Counterpoint* : it is true that providing users with the right incentives to reveal their actual costs of service degradation is complicated. However, with any prices that increase with the degree of congestion in the network, users will be induced to prioritize their traffic. Only users who value their traffic at least as much as the current price will transmit. If congestion remains unacceptably high, then the price was too low; conversely if capacity is unacceptably underutilized, the price was too high. Thus, through a process of experimentation and dynamic adjustment, the network can shape the price schedule so that users approximately reveal their valuations for uncongested service through their responses to the prices.
- why won't some non-pricing scheme be enough ? Administrative controls can be used to impose some appropriate notion of fairness, for example; or users can choose a traffic priority level which matches their requirements.
 - *Counterpoint* : who decides what is fair ? The network operator can; but according to a user-oriented objective, fairness should be determined collectively by the users. We might all agree that telesurgery is more important than email, but what about interactive video games versus email ? Also, every time a new application is developed it has to be slotted into the priority order, an increasingly complex process. Suppose the network simply supports priority levels and allows each user to choose their own level. Why wouldn't

they all choose the highest priority ? To guard against such abuses, there would have to be some penalty for “inappropriate” declarations, implying the need to define “appropriate” priority levels or to assign increasing charges to higher priorities, e.g. Bohn (1993). A user’s choice of priority level would then be based on economic considerations : balancing the benefits of higher priority against the costs and/or the penalties for inflating their application’s perceived priority level. Pricing represents the limiting case of a continuous spectrum of priorities.

- most users will want to know their charges in advance, and will not want to deal with prices that change during the lifetime of a typical connection.
 - *Counterpoint* : we are not advocating that all users must face usage-sensitive prices. Any user can choose not to face dynamic prices, even if their application is adaptive. They would then be charged according to some other pricing scheme, which should be co-ordinated with the usage-sensitive pricing mechanism. Or a user faced with dynamic prices can choose to ignore those prices by transmitting at their application’s natural information rate, and paying the resulting charges. Finally, in our scheme – and in any realistic pricing scheme – it would be possible for a user to set the **maximum** charge they are willing to pay, which is what is usually required for budgetary purposes.
- bits/bytes/cells are not the correct units to charge for – it’s information that users care about. Any scheme which proposes to look inside every data unit to determine how it relates to other data units is likely to be too complex to be justified. Also, lower-layer mechanisms (such as Ethernet collisions) or cell losses requiring higher-layer retransmissions make it difficult to predict how much “raw” data has to be transmitted to transfer a given amount of information. Should users be charged for retransmissions that they have no control over, or cells which are dropped by the network ?
 - *Counterpoint* : our scheme involves pricing for transport, not for content. The “importance” of a particular cell, and its relation to other cells, is a higher-layer issue determined by the application (or ultimately by the users). We are not proposing that the network be aware of these issues; on the contrary, the network view in our scheme is that it’s up to the users to decide how cells are used to transfer information. It is in general impossible to predict exactly how many cells are required to transmit a block of information, but again this is a higher-layer issue. The basic question is whether the users or the network should bear the uncertainty. If the network is expected to offer a “file transfer” service, the file transfer charge per megabyte could be computed by averaging over many such transfers. If the user is expected to pay for all transmitted cells, their application could for example define a maximum number of cells they are willing to transmit per megabyte of information, or a maximum amount they are willing to pay per megabyte of information actually transferred.

- dynamic pricing schemes are unworkable in practice due to the overheads involved in accounting and billing for usage on such a detailed level. In addition, a significant portion of the revenue raised is needed to defray the cost of doing dynamic pricing in the first place.
 - *Counterpoint* : the costs of dynamic pricing may outweigh the benefits for a particular implementation but we do not believe this is necessarily true for all dynamic pricing schemes. In particular, online pricing mechanisms may reduce the actual cost to an acceptable level; there is no reason to think that current billing and accounting costs in other industries, such as telephone or electricity networks, will necessarily apply to dynamic pricing in ATM networks. This concern can only be answered for each scheme individually, and should obviously be part of the overall decision on what usage-sensitive pricing scheme to implement, if any.
- dynamic pricing is impractical because users cannot respond to prices which are updated many times per second. If the update interval is increased to the minimum period in which users can respond, congestion can arise and disperse in between price updates, so that prices no longer influence user behaviour.
 - *Counterpoint* : our scheme assumes an intelligent network interface at price-sensitive user sites, so the processing necessary to respond to dynamic prices would be done automatically based on pre-programmed user preferences. Current ATM connection admission control schemes already assume enough user intelligence to be able to negotiate quality of service parameters, so our scheme adds a little more complexity rather than a new requirement. This software would play a similar role to current TCP implementations, which respond to network feedback by adjusting their traffic inputs, except that the feedback in our case is the current price.
- charging for cells transmitted fails to capture cases where the benefit of a transfer is with the receiver. If senders are charged for receiver-initiated transfers, we could see a drastic reduction in the number of open-access servers with a corresponding decrease in the value of using the network.
 - *Counterpoint* : we do not believe that associating the charge for a transmission with the sender constrains the actual flow of money in any way. It is easy to imagine multiparty connection protocols which initially negotiate each party's responsibility for the total charge, or "reverse-charges" servers which only transmit data once the receiver has indicated willingness to pay the resulting transmission costs.
- With any form of usage-sensitive pricing, it's the small users who will suffer the most. Rich users could behave as they want since they have the resources, and could effectively limit the network access of smaller users.
 - *Counterpoint*: with our scheme, if your application is flexible enough, your charges would be **zero**. Many opponents of usage-sensitive pricing seem to

believe that it inherently involves charging for every cell/bit/etc. However our scheme explicitly recognises that if there is no congestion in the network, the usage-sensitive price should be zero. Users who are flexible enough to wait for such periods can then transmit for free⁴. It would be a relatively simple matter to set a maximum price per cell of zero and let your network interface determine when to transmit – assuming your application can wait for the price to drop. As for rich users being able to afford to ignore dynamic prices, this is true under any pricing scheme and is not particular to dynamic pricing. We do not mean to dismiss income distribution problems as unimportant, but merely to say that network pricing (or non-pricing) is not the right venue for solving them.

- usage-sensitive pricing is just another way for network operators to make more money. Users will lose out as network operators maximise their profits.
 - *Counterpoint* : it's true that there is the potential for profiteering whenever prices are charged, especially when the conditions under which prices are set are not immediately accessible to ordinary users. But in a competitive environment, network operators have market incentives to keep their margins of revenue over actual cost as low as possible. This incentive is missing in the case of a monopoly provider or a cartel of price-fixing providers. But whether abuse is possible in this case depends on policy and regulatory decisions rather than on the specific pricing scheme.
- economics is important in network planning but has nothing to do with the technical operation of a network, whether public or private.
 - *Counterpoint* : economics has a lot to do with network operation ! Packet-switching was developed for computer communications because around 1970 it became more economical to use switching and routing to statistically multiplex several connections into one transmission link, rather than dedicating one circuit to each connection as in circuit-switching. Economics plays a role in formulating and solving decision problems in all types of network; current price schemes differ from ours in the frequency with which prices are updated. Our scheme simply moves this updating into “real-time” for those users who are able and willing to respond on that timescale.

5 CONCLUSIONS

We have presented an economic framework for adaptive users in ATM networks. Instead of the typical requirement for traffic descriptors in order to get performance guarantees, these flexible users can get loss guarantees if they adjust their traffic input rates in response to dynamic feedback from the network. This is the basis for recent proposals for

⁴If the price is never zero, the network is always congested and capacity expansion is indicated.

ABR service in ATM. Our framework takes these proposals one step further by explicitly defining how that feedback is generated by the network, and what form it takes. In our scheme the network associates a cost measure with the utilisation of network resources, announces a price which is based on the current cost, and price-sensitive users adjust their cell inputs based on this price and their own specification of how valuable network service is to them. While we address only the reactive control of adaptive users, our scheme could be part of a more comprehensive billing and accounting scheme to charge all users for network services.

What we propose is to give users incentives to consider the effects of their usage on other users. In a public network, where the users cannot be assumed to be cooperative, more traditional feedback schemes are not robust to user manipulation : it is relatively easy to program a host to ignore the feedback signals. Of course it would be just as easy to ignore price signals; but since users would be liable for charges they incurred, there is some incentive to respond.

We also address the problem of user service valuations, and allow for adaptive sources to have more demanding traffic than well-described sources. We have proposed a distributed iterative pricing algorithm and shown (by simulation) that it is possible to gain both network efficiency and economic efficiency by using pricing. In other words, the network actually carries more traffic and carries more important traffic from the users' point of view.

6 ACKNOWLEDGEMENTS

We thank Jeff MacKie-Mason for many useful discussions about this work, and the contributors to the *com-priv* Internet mailing list for their objections to usage-sensitive pricing.

7 REFERENCES

- ATM Forum (1993) 'ATM User-Network Interface Specification', Prentice Hall.
- Bohn, R., Braun, H. W., Claffy, K. and Wolff, S. (1993) 'Mitigating the coming Internet crunch: Multiple service levels via precedence,' Tech. rep., UCSD, San Diego Supercomputer Center, and NSF.
- COST (1993) 'Redundancy Reduction Techniques for Coding of Video Signals in Multi-Media Services', COST 211ter, Compendium ATM, Coding Procedures, Simulation Subgroup, pp 1-83.
- de Prycker, M. (1993) *Asynchronous Transfer Mode : Solution for Broadband ISDN*, 2nd Ed., Ellis Horwood.

Mackie-Mason, J. and Varian, H. (1994) 'Some FAQs about Usage-Based Pricing', available from URL <ftp://gopher.econ.lsa.umich.edu/pub/Papers/useFAQs.html>

Murphy, J. and Murphy, L. (1994) 'Bandwidth Allocation By Pricing In ATM Networks', *IFIP Transactions C : Communication Systems*, No. C-24, pp. 333-351, available from URL <http://www.eeng.dcu.ie/~murphyj/band-price/band-price.html>

Murphy, J., Murphy, L. and Posner, E. C. (1994) 'Distributed Pricing For Embedded ATM Networks', *Proc. International Teletraffic Congress ITC-14*, Antibes, France, pp. 1053-1063, available from URL <http://www.eeng.dcu.ie/~murphyj/dist-price/dist-price.html>

Murphy, J. and Teahan, J. (1994) 'Video Source Models for ATM Networks', *Eleventh UK IEE Teletraffic Symp.*, Cambridge, UK, available from URL <http://www.eeng.dcu.ie/~murphyj/video/video.html>

Murphy, L. and Murphy, J. (1995) 'Pricing for ATM Network Efficiency', *Proc. 3rd International Conference on Telecommunication Systems Modelling and Analysis*, Nashville, TN, pp. 349-356, available from URL <http://www.eeng.dcu.ie/~murphyj/atm-price/atm-price.html>

Murphy, J. (1996) *Resource Allocation In ATM Networks*, Ph.D. thesis, School of Electronic Engineering, Dublin City University, Ireland, available from URL <file://ftp.eeng.dcu.ie/pub/tele-communications/murphy/thesis/thesis.ps>

Ramakrishnan, K. K. and Newman, P. (1995) 'Integration of Rate and Credit Schemes for ATM Flow Control', *IEEE Network*, p. 49-56.

SES (1992) *SES/workbench Reference Manual*, Release 2.1.

8 BIOGRAPHIES

Liam Murphy is an Assistant Professor in the Department of Computer Science and Engineering at Auburn University. He received his PhD in Electrical Engineering from the University of California at Berkeley in 1992. His current research interests are in the areas of resource allocation and congestion control in integrated-services networks, and multimedia networking. He is a member of the IEEE.

John Murphy is a lecturer in the School of Electronic Engineering at Dublin City University, where he received his PhD in 1996. He has worked with AT&T, Telecom Eireann and JPL on various teletraffic problems. His current research interests are in the areas of resource allocation in ATM networks and in modelling and simulation of high speed networks. He is a member of the IEEE.

PART SIX

Models of ATM Switches

Geometrical bounds on an output stream of a queue in a model of a two-stages interconnection network : application to the dimensioning problem

L. Truffet

Laboratoire MASI-CNRS UA 818

Universite Paris VI Institut Blaise Pascal 4, Place Jussieu F-75252 Paris Cedex 05 FRANCE

Tel 44 27 61 93, e-mail : truffet@masi.ibp.fr

Abstract

On a discrete time Markovian model of a two-stage interconnection network we develop methods to find stochastic bounds on the number of lost cells at the second stage. We deduce bounds on the lost rate. These methods are based on comparison results of stochastic processes, Veinott's criterion and lumpability of transition matrices. We study the output process of a discrete time $Geom^X/D/1$ queue at the first stage and we compute geometrical bounds on the output process of this $Geom^X/D/1$ queue.

Keywords

Multi-Stage Interconnection Network, Discrete time Markovian models, Strong Ordering, Veinott's Criterion, Lumpability.

1 INTRODUCTION

Most of the telecommunications systems are composed of interconnected nodes. The messages routed from a source to a destination through the interconnection network of nodes are generally cut into numbered frames also called packets or cells (in the context of the Asynchronous Transfert Mode (ATM)). Each node is a system which switches packets of messages from its input to the desired output. Roughly speaking a node is a black box composed of a commutation function (insures the routing of a packet from the input to the output of the node) and a capacity to store packets waiting the commutation. If a packet arrives at a full node it is lost. One of the most important problem is to determine the capacity of a node such that very few packets will be lost. This is called the dimensioning problem. To address this problem we have to compute the lost rate (i.e, the ratio of the mean number of packets lost by the mean number of packets arriving at the considered node).

The general context of our analysis is the one recommended by the CCITT which is ATM. For this transfer mode the packets have a fixed length (53 bytes) to insure a quick commutation in a node. Due to the fixed length of the cells and the fixed commutation duration the time is discretized into unity called slot. Knowing the fact that the behavior of a node depends of the behaviors of the other nodes connected to its inputs and also depends of the behaviors of the sources we first want to study only the behavior of a node submitted to an input traffic. This traffic is modeled by a stochastic process. This process represents the resulting behaviors of the sources and the other nodes of the network.

Performance of a node or switch depends of its input traffic and its architecture. One of the most studied architecture for a node is the Multi-stages Interconnection Network (MIN). It seems that the choice of such architecture is made both by the industrials and the researchers. This is due to the fact that the realization of MIN has a low cost and performance measures interesting. Such a switch is composed of switching elements with very few inputs and outputs (to insure a very fast commutation). These switching elements have a commutation function and capacity to store cells waiting to be switched. The switching elements are connected by an interconnection network.

The final aim is to propose numerical methods with low complexity to address the dimensioning problem for MINs. This will imply that these methods allow a designer of switches to choose the best possible configuration for a switch submitted to several kinds of input traffic (i.e, Bernoulli traffic, Bursty Geometric or ON/OFF traffic, and so on). In general a MIN is represented by a feed-forward queuing network and choosing a configuration means to choose the number of queues and their capacities and also the routing network between the queues. Let us also note that it is very important to be able to observe the behavior of a switch submitted to several kinds of traffic with different variance for the same mean number of arrivals because the lost rate depends of this variance.

In this paper we are interested in the dimensioning problem of MINs. The performance criterion considered is the loss rate. This is a classical problem in telecommunication networks but the imposed lost rate of 10^{-9} in high speed networks as ATM networks is a new difficulty in the performance evaluation of these kind of networks compared with the dimensioning problem of telephone networks with an imposed rate of rejected (or lost) calls of about 10^{-2} . Discrete time Markovian models of MINs have state space with a very high number of states in practice and these kind of networks have no analytical solution, no product form solution for instance. Because of the number of states in these models the algorithmic complexity does not allow the computation of the exact solution for lost rate in particular. The use of approximation techniques to reduce states space is a natural way to address the dimensioning problem but pure approximation techniques must be validated using rare or very rare event simulation techniques which are generally very expensive techniques. So what we propose in this paper is to develop special approximation techniques to obtain bounds which permit us to avoid validations by simulation. These bounds must be easy to compute (i.e., the complexity must be much lower than the complexity of the computation of the exact solution). The aim of this work is to present the basic methods and the basic mathematical results to address the dimensioning problem of a very simple model of MIN. This model is a Markovian model of a two-stages interconnection network with Bernoulli input traffic. Of course this model is not realistic when modeling ATM switch. This is due to the fact

that the Bernoulli traffic is not realistic but we focus our attention on the methodologies and we will give some tracks how to modify the basic methods for more general input traffics.

The starting point to obtain bounds is to use some structural properties of the model. The basic property we use in this work is that the number of lost cells at each slot t in a queue of a switching element is an increasing function of its input processes. Now what we want is to bound the output process of a queue by a very simple renewal process. That is why we propose methods for finding geometrical bounds on the output stream of a queue using comparison results on random variables and processes. We use the strong comparison [Sto76] because it is generated by increasing functions and the number of lost cells is an increasing function of the input processes. The second idea is to reduce the state space (note that this is generally done by other techniques) that is why we investigate results on lumpability [JJ60], [RS91].

The paper is organized as follows. The section 2 deals with the mathematical results useful for obtaining bounds on the loss rate. The main result concerns the comparison of Markov processes. The section 3 is devoted to the model description of the interconnection network. Roughly speaking this system could be considered as the first two stages of a Clos interconnection network [Clo53]. In section 4 we present methods based on lumpability results and in section 5 we present a method based on Veinott's Criterion to obtain an upper bound on the output stream of $Geom^X/D/1/FCFS/C$ queue with service duration equal to 1. The section 6 is devoted to the study of numerical examples of the bounding methodologies and a discussion on these numerical results. Then we conclude in the last section.

2 MATHEMATICAL RESULTS

In this section we present the main central result (see proposition 2.3) concerning the comparison of discrete time Markov processes in the sense of the strong ordering. But before that we need to introduce some definitions and properties.

The strong ordering is the basic notion of this paper. This ordering is generated by non-decreasing functions.

Definition 2.1 (Strong Ordering) *Let $k > 0$ be an integer. Let \vec{X} and \vec{Y} be two \mathbb{R}^k -valued random variables. We say that \vec{X} is lower than \vec{Y} in the sense of the strong ordering iff :*
for all functions $f : \mathbb{R}^k \rightarrow \mathbb{R}$ nondecreasing in the sense of the componentwise ordering on \mathbb{R}^k the inequality

$$Ef(\vec{X}) \leq Ef(\vec{Y})$$

holds, provided that the expectations exist.

The strong ordering has many properties and the reader is referred to [Sto76] for more details on this subject but here we need to mention the following property which will be used to simplify the study.

Proposition 2.1 (Comparing independent variables) *Let us consider two vectors of independent variables, respectively denoted by $X = (X_1, \dots, X_n)$ and $Y = (Y_1, \dots, Y_n)$. We then have the following result :*

$$X \leq Y \text{ iff } \forall i \ X_i \leq_{st} Y_i$$

Let us notice that the previous proposition could be generalized to the case where X_i (resp. Y_i) is a random vector, for all $i = 1, \dots, n$.

To compare a vector of random variables in the sense of \leq_{st} , we will use a sufficient condition called Veinott's Criterion (1965). We borrow the definition from [Sto76] suited to our case.

Proposition 2.2 (Veinott's Criterion) *Let $U = (U_0, \dots, U_t)$ and $V = (V_0, \dots, V_t)$ be two vectors of random variables taking their values in \mathcal{E} .*

If

$$U_0 \leq_{st} V_0 \tag{1a}$$

and

$$\begin{aligned} & \forall j = 1, \dots, t \ \forall (x, u, v) \in \{0, \dots, M\} \times \{0, \dots, M\}^j \times \{0, \dots, M\}^j, \ u \leq v \\ & Pr(U_j \geq x | U_0 = u_0, \dots, U_{j-1} = u_{j-1}) \leq Pr(V_j \geq x | V_0 = v_0, \dots, V_{j-1} = v_{j-1}) \end{aligned} \tag{1b}$$

noting that $u = (u_0, \dots, u_{j-1})$ and $v = (v_0, \dots, v_{j-1})$

then $U \leq_{st} V$

The above criterion is very useful because it is taking into account the fact that the random vectors may have correlated components (which is the case when studying Markovian processes).

Because we are studying processes with state space $\mathcal{E} = \{0, \dots, M\}$, we give a specified version of definition 2.1 in this particular case. Noting that the set of nondecreasing functions on $\mathcal{E} = \{0, \dots, M\}$ is generated by the functions $1_{\{x \geq k\}}, k = 0, \dots, M$ we can express the \leq_{st} -comparison as follows.

Let be X and Y two random variables taking their values in $\{0, \dots, M\}$. Let p (resp. q) be the row vector of distribution of X (resp. Y). We say that X is lower than Y in the sense of \leq_{st} ordering and we denote $X \leq_{st} Y$ (or equivalently $p \leq_{st} q$) iff

$$p K_{st} \leq q K_{st} \text{ componentwise} \tag{2}$$

with K_{st} the $(M+1) \times (M+1)$ following matrix :

$$K_{st} = \begin{pmatrix} 1 & 0 & 0 & \dots & 0 \\ 1 & 1 & 0 & \dots & 0 \\ 1 & 1 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & 1 & 1 & 1 \end{pmatrix} \tag{3}$$

For instance if $M = 2$ then

$$K_{st} = \begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{pmatrix}$$

If $p = (0.1, 0.3, 0.6)$ and $q = (0.7, 0.2, 0.1)$, then computing $p K_{st} = (1, 0.9, 0.6)$ and $q K_{st} = (1, 0.3, 0.1)$ we see that $p \geq_{st} q$.

With the above mathematical tools we are now able to write the central result on comparison of Markovian processes in the following proposition.

Proposition 2.3 (Comparing Markov Processes) *Let $\{X_t, t \geq 0\}$ and $\{Y_t, t \geq 0\}$ be two Markovian processes taking their values in \mathcal{E} . These processes are also denoted by (p_0, P) and (q_0, Q) , respectively. The row vector p_0 (resp. q_0) denotes the distribution vector of X_0 (resp. Y_0), P (resp. Q) denotes the transition matrix of $\{X_t, t \geq 0\}$ (resp. $\{Y_t, t \geq 0\}$). If*

$$p_0 \leq_{st} q_0 \tag{4a}$$

$$\text{and} \quad P K_{st} \leq Q K_{st} \text{ term by term comparison} \tag{4b}$$

and $A = P$ or $A = Q$ is monotone ie :

$$\forall i \leq j \quad A_{i,\cdot} \leq_{st} A_{j,\cdot} \tag{4c}$$

where $A_{i,\cdot}$ is the i^{th} row of matrix A ,
then

$$\forall t \quad X_t \leq_{st} Y_t$$

moreover

$$\forall t \quad (X_t, \dots, X_0) \leq_{st} (Y_t, \dots, Y_0)$$

-Proof :

For the point by point comparison see [Kei77]. For the vectorial comparison we just have to note that Veinott's Criterion for Markovian processes is equivalent to (4b). This vectorial comparison could also be obtained using coupling argument (see Doisy [Doi92]) \square

3 MODEL DESCRIPTION

In this section we present a simple model of a switch architecture submitted to Bernoulli input traffic. This could be represented a very simple and not very realistic model (due to the assumption concerning the input traffic) ATM switch.

The model of the ATM switch studied here is the same as that described in [BM93], [Bey93] where most of assumptions made are the same as in [KHM87]. After describing it we present

the evolution equations and the fundamental properties which induce the methods proposed here for bounding the loss rate. In the last point we present fundamental assumptions for using our bounding methodologies.

3.1 The model

Consider a two-stage interconnection switch with N entries and M outputs (see Fig. 2). We assume that this system operates in a discrete way at each time slot and we make our analyze at the cell level.

We make the hypothesis that departures occur before arrivals of cells.

The arrival processes at each entry are geometrically distributed with parameter p . Arrival processes at two different inputs are assumed to be independent. Let us notice that in the ATM context this assumption does not allow to treat the problem of the bursty traffics. This will be discussed at the end of this paper.

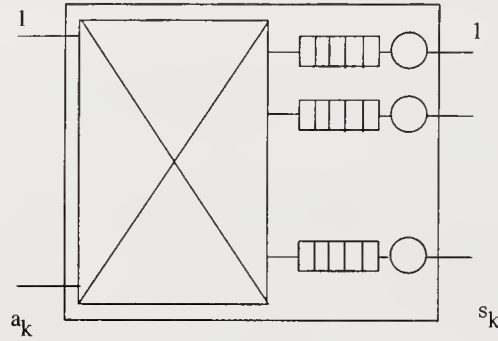


Figure 1: Switching Element of stage k

We assume that each stage k , $k \in \{1, 2\}$, is composed of E_k non-blocking identical switching element. A switching element at stage k , $k \in \{1, 2\}$ (see Fig. 1) has a_k inputs and s_k outputs. Each output is a queue with service duration 1 (slot), service discipline *FCFS* and a finite capacity M_k . This queue is denoted by $G^X/D/1/FCFS/M_k$ and its service time is equal to 1. Let us notice that for $k = 1$, the arrival process at a queue of the first stage is a bulk geometrical process and the size of the batch arrival is less or equal to a_1 (recalling that a_1 is the number of inputs of a switching element at stage 1).

Let us denote (i, j, k) (resp. (o, j, k)) the input i (resp. output o) of the switching element j at stage k , $k \in \{1, 2\}$.

The commutation function of a switching element connects all input ports to all output ports. This function is supposed to be uniform, i.e the probability that a cell arriving at input (i, j, k) being switched to output (o, j, k)

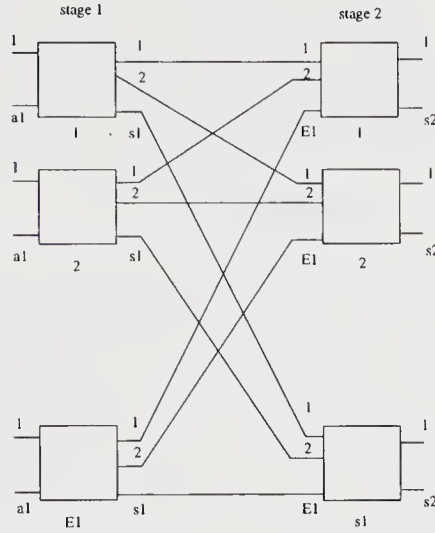


Figure 2: A Two-stage Interconnection Network

is the same for all o and then is equal to $\frac{1}{s_k}$, independent of the state of the switch (i.e, the number of cells in each queue of the switch).

The interconnection network is modeled by the following bijection

$$C : \begin{array}{l} \{1, \dots, s_{k-1}\} \times \{1, \dots, E_{k-1}\} \times \{1\} \longrightarrow \{1, \dots, a_k\} \times \{1, \dots, E_k\} \times \{2\} \\ (o, j, k) \longmapsto (i, j, k+1) = (j, o, k+1) \end{array}$$

whose inverse is

$$C^{-1} : (i, j, k) \longmapsto (o, j, k-1) = (j, i, k-1)$$

As an immediate consequence on the design of such a MIN we have :

$$a_1 = E_2 \quad (5)$$

3.2 Evolution equations

Let us denote by $N_{o,j}^k(t)$, the random variable number of cells of queue o of switching element j at stage k .

Because of the previous assumptions made the stochastic process $\{S(t), t \in \mathbb{N}\}$ is Markovian, where \mathbb{N} is the set of integers and :

$$S(t) = (N_{o,j}^k(t))_{\{k=1,2; j=1 \dots E_k; o=1 \dots s_k\}}$$

The evolution equations of such system are :

$$\forall o, j, k \quad \begin{cases} N_{o,j}^k(t) = 0 \\ N_{o,j}^k(t+1) = \min(M_k, (N_{o,j}^k(t) - 1)^+ + A_{o,j}^k(t)) \end{cases} \quad t \geq 0 \quad (6)$$

where $(x)^+ = \max(0, x)$, $A_{o,j}^k(t)$ is the number of cells arriving to the output o of j at stage k between $]t, t+1]$. $A_{o,j}^k(t)$ is defined as follows :

$$A_{o,j}^k(t) = \sum_{i=1}^{a_k} I_{i,j}^{k-1}(t) (e_o | u_{i,j}^k(t)) \quad (7)$$

where $(|)$ denotes the canonical scalar product in IR^n . $I_{i,j}^{k-1}(t) = 1_{\{N_{i,j}^{k-1}(t) > 0\}}$, by definition of the connection between stages. Note that if $k = 1$ the variables $I_{i,j}^{k-1}(t)$ are iid Bernoulli distributed with parameter p .

The random variables $u_{i,j}^k(t)$ represent the commutation function of a switching element and take their values $e_1, \dots, e_o, \dots, e_{s_k}$ where $e_o, o = 1, \dots, s_k$ is a row vector with s_k components all equal to 0 except the o^{th} which is equal to 1. The event $u_{i,j}^k(t) = e_o$ (or equivalently the event $(u_{i,j}^k(t) | e_o) = 1$) means that cell arriving at i chose the output o of the switching element j at stage k .

3.3 Fundamental properties

The following propositions give the fundamental properties of the model used in finding bounds on the lost rate : at each slot t the number of cells in a queue and the number of lost cells at the same queue are increasing functions of the input processes.

Proposition 3.1 (The number of cells is a nondecreasing function of the inputs) *For all $k \geq 0$, for all $t \geq 0$ we have the following result.*

For all o, j , $N_{o,j}^k(t)$ is a nondecreasing function of $((I_{i,j}^{k-1}(t-1), \dots, I_{i,j}^{k-1}(0)); (c_{i,j}^k(t), \dots, c_{i,j}^k(0)))$

where :

$$\forall t' \quad I_{i,j}^{k-1}(t') = (I_{i,j}^{k-1}(t'))_{i=1, \dots, a_k}$$

and

$$\forall t' \quad c_{i,j}^k(t') = ((e_o | u_{i,j}^k(t')))_{i=1, \dots, a_k}$$

-Proof :

Using relations (6) and (7), the result is proved by induction on t . \square

Moreover if $\Pi_{o,j}^k(t)$ denotes the number of lost cells at queue o of switching element j at stage k and at instant t , then we have the following result :

Proposition 3.2 (The number of lost cells is a nondecreasing function of inputs) *For all $k \geq 0$, for all $t \geq 0$ we have the following result.*

For all o, j , $\Pi_{o,j}^k(t)$ is a nondecreasing function of $((I_{i,j}^{k-1}(t), \dots, I_{i,j}^{k-1}(0)); (c_{i,j}^k(t), \dots, c_{i,j}^k(0)))$

where :

$$\forall t' \quad I_{i,j}^{k-1}(t') = (I_{i,j}^{k-1}(t'))_{i=1, \dots, a_k}$$

and

$$\forall t' \quad c_{i,j}^k(t') = ((e_o | u_{i,j}^k(t'))_{i=1, \dots, a_k}$$

-Proof :

We just have to note that the evolution equations of the number of lost cells are

$$\begin{cases} \Pi_{o,j}^k(0) &= 0 \\ \Pi_{o,j}^k(t) &= ((N_{o,j}^k(t) - 1)^+ + A_{o,j}^k(t) - M_k)^+ \end{cases} \quad (8)$$

and then use the previous result of proposition 3.2. \square

Finally we have to mention the following remark :

By assumption the random variables $\{I_{i,j}^0(t)\}_{1 \leq i \leq a_1, 1 \leq j \leq E_1}$ are independent for all $t \geq 0$. Then due to the interconnection network between stage 1 and stage 2 the random variables $\{I_{i,j}^1(t)\}_{1 \leq i \leq a_2}$ are also independent.

This last remark means that if the entries of a switch are independent then the input ports of a switching element of the second stage are also independent. This is another restrictive case for applying our bounding method of a switch model. But this means for instance that a Delta network with independent inputs have the same property at each stage : the inputs of each switching element are independent.

3.4 The dimensioning problem

Define the lost rate at stage k , $k \in \{1, 2\}$ by

$$\pi(k) = \lim_{t \rightarrow +\infty} \frac{E\Pi_{o,j}^k(t)}{EA_{o,j}^k(t)} \quad (9)$$

then, the dimensioning problem of the designer of switch architecture can be characterized by the following iterative algorithm using an exact or a bounding methodology :

DimSwitch

- step 1 : Define the Switch $(E_1, a_1, s_1, a_2, s_2)$
- step 2 : Put capacity of queues M_k , $k = 1, 2$
- step 3 : Compute exact (if possible) or lower and upper bounds on $\pi(k)$, $k = 1, 2$
- step 4 : If $\pi(1)$ and $\pi(2)$ are acceptable then Goto step 5
- Else, goto step 2
- step 5 : If designer decide it is OK Stop Else Goto step 1

This procedure means that the designer has to fix first an architecture (step 1). Then he has to propose some capacities for the queues (step 2). For the whole configuration one has to estimate (or bound) the loss rate (step 3). If the loss rate is acceptable (i.e, less or equal to a given value) then the designer can decide to stop the procedure or to explore new architecture (i.e, restart a whole procedure). Let us remark that the decision to stop the design process is generally due to the calculus of an economical cost of the implementation of designed switch.

3.5 Difficulty of the problem

Because of assumptions on input processes and switching function, it has been shown (see [Bey93], []) that the behaviors of the queues of the first stage are identical. These behaviors are represented by the processes $\{N_{o,j}^1(t), t \in \mathbb{N}\}$ which are identical Markovian processes whose transition matrix denoted by T_1 is defined as

$$T_1 = \begin{cases} \begin{pmatrix} b_0(p) & b_1(p) & b_2(p) & \cdots & b_{M_1}(p) + \cdots + b_{a_1}(p) \\ b_0(p) & b_1(p) & b_2(p) & \cdots & b_{M_1}(p) + \cdots + b_{a_1}(p) \\ 0 & b_0(p) & b_1(p) & \cdots & b_{M_1-1}(p) + \cdots + b_{a_1}(p) \\ & \ddots & \ddots & & \\ 0 & \cdots & 0 & b_0(p) & b_1(p) + \cdots + b_{a_1}(p) \end{pmatrix} & \text{if } a_1 > M_1 \\ \begin{pmatrix} b_0(p) & b_1(p) & b_2(p) & \cdots & b_{a_1}(p) & 0 & \cdots & 0 \\ b_0(p) & b_1(p) & b_2(p) & \cdots & b_{a_1}(p) & 0 & \cdots & 0 \\ 0 & b_0(p) & b_1(p) & \cdots & b_{a_1}(p) & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & & & & & \\ 0 & \cdots & 0 & b_0(p) & b_1(p) & b_2(p) & \cdots & b_{a_1}(p) \\ \vdots & & & \ddots & b_0(p) & b_1(p) & \cdots & b_{a_1-1}(p) + b_{a_1}(p) \\ \vdots & & & & \ddots & b_0(p) & b_1(p) & b_2(p) + \cdots + b_{a_1}(p) \\ 0 & \cdots & & & \cdots & 0 & b_0(p) & b_1(p) + \cdots + b_{a_1}(p) \end{pmatrix} & \text{otherwise} \end{cases} \quad (10)$$

for all o, j , and $t > 0$ where,

$$b_i(p) = Pr(A_{o,j}^0(t) = i) = \binom{a_1}{i} \left(\frac{p}{s_1}\right)^i \left(1 - \frac{p}{s_1}\right)^{a_1-i} \quad (11)$$

is the probability that i cells arrive at queue $(o, j, 1)$.

We note that $\pi(1)$ is computable (we only have to compute the steady-state probability vector of T_1). The only problem still remaining is the exact computation of $\pi(2)$. To compute $\pi(2)$ we only have to study a queue at the second stage and the upstream queues of the first connected to it. This problem is an $O(((M_1 + 1)^{E_1} \times (M_2 + 1))^3)$ complexity problem because this kind of networks has no analytical solution (based for instance on a product form result).

So, the major problem we have is a complexity problem. It means that for a "small" switch with a few number of queues with very low capacities it is possible to obtain the exact value of $\pi(2)$. But this implies that we cannot explore a lot of switch configurations.

However noticing that (see [Bey93]) :

$$\forall o, j \lim_{t \rightarrow +\infty} E(A_{o,j}^1(t)) = \frac{a_2}{s_2} p(1 - \pi(1)) \quad (12)$$

we only have to focus our attention on the computation of $\lim_{t \rightarrow +\infty} E(\Pi_{o,j}^2(t))$. The result of proposition 3.2 suggests the following fundamental remarks to develop bounding methodologies : if the input processes of a switching element at stage 2 are Bernoulli then $\pi(2)$ is easy to compute because of the increasing function property of the number of lost cells at each slot t we want to use the strong ordering,

because of the variables $I_{i,j}^k(t) = 1_{\{N_{j,i}^{k-1}(t) > 0\}}$ (let us recall that the output stream of a queue (o, j, k) is the stochastic process $\{1_{\{N_{o,j}^k(t) > 0\}}, t \in \mathbb{N}\}$) we want to use lumpability results [JJ60] because the input stream of a switching element at stage 2 are independent and the result of the proposition 2.1 we only have to bound (in the sense of \leq_{st}) the output stream of a queue of stage 1 by Bernoulli processes (also called geometrical processes).

Finally we just have to focus our attention to the resolution of the following problem. Let us consider an homogeneous and irreducible Markov chain $\{X_t, t \in \mathbb{N}\}$ with state space $\mathcal{E} = \{0, \dots, M\}$ which will be denoted from now by (τ_0, T_X) (where τ_0 is the initial condition and T_X the transition matrix) and a process $\{Y_t, t \in \mathbb{N}\}$ such that for all $t \geq 0$, Y_{t+1} is an increasing function of the vector $(1_{\{X_t > 0\}}, \dots, 1_{\{X_0 > 0\}})$. We want to find geometrical lower (resp. upper) bounds $Ginf$ (resp. $Gsup$) such that :

$$\forall t \geq 0 (Ginf(t), \dots, Ginf(0)) \leq_{st} (1_{\{X_t > 0\}}, \dots, 1_{\{X_0 > 0\}}) \leq_{st} (Gsup(t), \dots, Gsup(0)) \quad (13)$$

4 BOUNDS BY AGGREGATION

The aim of this section is to find lower (resp. upper) Markovian processes X_{inf}^{sl} (resp. X_{sup}^{sl}) strongly lumpable according to the partition $\mathcal{B} = (B(0), B(1))$ with $B(0) = \{0\}$ and $B(1) = \{1, \dots, M\}$ such that processes $Ginf = \{1_{\{X_{inf}^{sl}(t) > 0\}}, t \geq 0\}$ and $Gsup = \{1_{\{X_{sup}^{sl}(t) > 0\}}, t \geq 0\}$ are geometrical delayed processes (see [Ros82]) and satisfy (13).

First we give the definition of strong lumpability in our special case.

Definition 4.1 (Strong Lumpability [JJ60]) A Markov chain (τ_0, T_X) is strongly lumpable according to \mathcal{B} iff

$$\forall k \in B(1) T_X(k, B(0)) = cste = T_X(0, 0) \quad (14)$$

Theorem 4.1 (Main Result) (τ_0, T_X) is a Markovian process such that $\{1_{\{X_t > 0\}}, t \geq 0\}$ has a geometrical lower (resp. upper) bound of parameter ρ_{inf}^{sl} (resp. ρ_{sup}^{sl}) with

$$\rho_{inf}^{sl} = \min_{k \in B(1)} \sum_{j=1}^M T_X(k, j) \quad (15)$$

and

$$\rho_{sup}^{sl} = \max_{k \in B(1)} \sum_{j=1}^M T_X(k, j) \quad (16)$$

-Proof :

Because of the definitions of ρ_{inf}^{sl} and ρ_{sup}^{sl} we note that $T_{inf}^{sl}K_{st} \leq T_X K_{st} \leq T_{sup}^{sl}K_{st}$ with

$$T_{inf}^{sl} = \begin{pmatrix} 1 - \rho_{inf}^{sl} & \rho_{inf}^{sl} & 0 & \cdots & 0 \\ 1 - \rho_{inf}^{sl} & \rho_{inf}^{sl} & 0 & \cdots & 0 \\ 1 - \rho_{inf}^{sl} & \rho_{inf}^{sl} & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 - \rho_{inf}^{sl} & \rho_{inf}^{sl} & 0 & \cdots & 0 \end{pmatrix} \quad (17a)$$

and

$$T_{sup}^{sl} = \begin{pmatrix} 1 - \rho_{sup}^{sl} & 0 & \cdots & 0 & \rho_{sup}^{sl} \\ 1 - \rho_{sup}^{sl} & 0 & \cdots & 0 & \rho_{sup}^{sl} \\ 1 - \rho_{sup}^{sl} & 0 & \cdots & 0 & \rho_{sup}^{sl} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 - \rho_{sup}^{sl} & 0 & \cdots & 0 & \rho_{sup}^{sl} \end{pmatrix} \quad (17b)$$

We see that the processes (τ_0, T_{inf}^{sl}) and (τ_0, T_{sup}^{sl}) are Markovian monotone and strongly lumpable according to \mathcal{B} . So we can apply result of proposition 2.3.

To complete the proof, note that a $\{0,1\}$ -valued Markov process with a transition matrix $\begin{pmatrix} 1 - \rho & \rho \\ 1 - \rho & \rho \end{pmatrix}$ is a delayed geometrical process. \square

5 BOUNDS USING VEINOTT'S CRITERION

Another way to obtain bounds is to use directly Veinott's Criterion. We only develop the method to obtain an upper bound because of the duality of the problem. Let us consider a Markov chain (τ_0, T_X) with state space \mathcal{E} . Our aim is to find a geometrical delayed upper bound $Gsup$ such that :

$$Gsup(0) = 1 \quad (18a)$$

$$\forall t \geq 1 \ Gsup(t) \text{ iid} \quad (18b)$$

and

$$\forall t(1_{\{X_t > 0\}}, \dots, 1_{\{X_0 > 0\}}) \leq_{st} (Gsup(t), \dots, Gsup(0)) \quad (18c)$$

Definition 5.1 (Possible Sequence) For a fixed initial condition τ_0 , the sequence $B(u_0), \dots, B(u_t)$ with $\forall i, u_i \in \{0, 1\}$ and $\forall i B(u_i) \in \mathcal{B}$ is possible iff

$$Pr_{\tau_0}(X_t \in B(u_t), \dots, X_0 \in B(u_0)) > 0 \quad (19)$$

Theorem 5.1 The parameter ρ_{sup}^{vc} of the geometrical upper bound is such that

$$\rho_{sup}^{vc} = \min_{\tau_0} \sup_{t > 0, B(u_t), \dots, B(u_0)} Pr_{\tau_0}(X_t \in B(1) | X_0 \in B(u_0), \dots, X_{t-1} \in B(u_{t-1})) \quad (20)$$

where $B(u_0), \dots, B(u_{t-1})$ is a possible sequence for the initial condition τ_0 .

-Proof :

For a fixed initial condition τ_0 a sufficient condition for satisfying (18a)-(18c) for all $t > 0$, using Veinott's Criterion (1a-1b) could be written in a simple manner (because $Gsup(t)$ are iid), that is :

$$\forall t \forall (x, u) \in \{0, 1\} \times \{0, 1\}^t$$

$$Pr_{\tau_0}(1_{\{X_t > 0\}} \geq x | 1_{\{X_{t-1} > 0\}} = u_{t-1}, \dots, 1_{\{X_0 > 0\}} = u_0) \leq Pr(Gsup(t) \geq x) \quad (21)$$

the case $x = 0$ is trivially satisfied. Then the result is obtained by definition of the function sup and using the fact that we want the smallest possible geometrical bound. \square

Now we have to give condition of existence and to give a way for computing this bound. If $T_X^+ = [T_X(i, j)]_{(i,j) \in B(1) \times B(1)}$, then we have the following result.

Theorem 5.2 (Main result) *If $T_X^+ = [T_X(i, j)]_{(i,j) \in B(1) \times B(1)}$ is a positive matrix such that the following assumptions are true*

$A_1 : T_X^+$ has no null row vector or T_X^+ is invertible

$A_2 : T_X^+$ is diagonalizable in \mathbb{C} , the set of complex numbers, or irreducible

then the parameter ρ_{sup}^{vc} exists and is computable using the following algorithm (where $r(T_X^+)$ is the maximum eigenvalue of T_X^+)

Begin

```

 $\omega = \frac{1}{\sum_{k=1}^M T_X(0, k)} (T_X(0, 1), \dots, T_X(0, M))$ 
 $max = \max(r(T_X^+), \|\omega T_X^+\|_1)$  (* where :  $\|(x_1, \dots, x_n)\|_1 = \sum_{i=1}^n x_i$  *)
While (1) do (* loop *)
     $\omega = \frac{\omega T_X^+}{\|\omega T_X^+\|_1}$ 
     $max = \max(max, \|\omega T_X^+\|_1)$ 
enddo (* end loop *)
```

End.

which converges.

-Proof :

The fact that $r(T_X^+)$ exists and is associated to a vector of distribution is due to the Perron-Frobenius-Gantmacher [Gan64] theory.

Noticing that $1_{\{X_i > 0\}} = u_i$, $u_i \in \{0, 1\}$ has exactly the same meaning than $X_i \in B(u_i)$, $u_i \in \{0, 1\}$, we use result in Rubino et al [RS91] to write that :

$$Pr_{\tau_0}(X_t \in B(1) | X_0 \in B(u_0), \dots, X_{t-1} \in B(u_{t-1})) = Pr_{\beta}(X_1 \in B(1))$$

with $\beta = f(\tau_0, B(u_0), \dots, B(u_{t-1}))$, where f is defined for all possible sequence by

$$\begin{cases} f(\tau_0, B(u_0)) &= \tau_0^{B(u_0)} \\ f(\tau_0, B(u_0), \dots, B(u_k)) &= (f(\tau_0, B(u_0), \dots, B(u_{k-1}))T_X)^{B(u_k)} \end{cases} \quad (22)$$

where α^C , $C \in \mathcal{B}$ denotes the row vector with $Card(C)$ components defined for any distribution vector α such that $\sum_{k \in C} \alpha(k) \neq 0$ by :

$$\forall i \quad \alpha^C(i) = \begin{cases} \frac{\alpha(i)}{\sum_{k \in C} \alpha(k)} & \text{if } i \in C \\ 0 & \text{otherwise} \end{cases} \quad (23)$$

using the fact that $f(\tau_0, B(u_0), \dots, B(0), B(u_i), \dots, B(u_k)) = f(1^{B(0)}, B(u_i), \dots, B(u_k))$, where 1 denotes the row vector with all its components equal to 1, we only have to consider two infinite sequences : $\mathcal{S}(0) = B(0), B(1), \dots, B(1), \dots$ and $\mathcal{S}(1) = B(1), B(1), \dots, B(1), \dots$. These two special sequences mean that in fact we just have to study the time spent in the partition $B(1)$ when at instant 0 we were in $B(0)$ or already in $B(1)$.

Then noticing that

$$f(\tau_0, B(0), B(1), \dots, B(1)) = f(1^{B(0)}, B(1), \dots, B(1)),$$

$$f((0, v_1), B(1), \dots, B(1)) = (0, v_1), \text{ with } v_1 \text{ the probability vector associated to the eigenvalue } r(T_X^+),$$

$$((0, v)T_X)^{B(1)} = \frac{vT_X^+}{\|vT_X^+\|_1}$$

$$\text{and the series defined by } \begin{cases} \omega_0 > 0 & \|\omega_0\|_1 = 1 \\ \omega_{t+1} &= \frac{\omega_t T_X^+}{\|\omega_t T_X^+\|_1} \end{cases} \text{ is such that } \lim_{t \rightarrow +\infty} \|\omega_t T_X^+\|_1 \text{ converges to } r(T_X^+),$$

the result is obtained. \square

6 NUMERICAL RESULTS

In this section we give numerical results concerning the dimensioning problem applied to a two-stage interconnection network to obtain a loss rate less or equal to 10^{-9} .

The configuration of such a switch is completely defined by the 4-tuple (E_1, a_1, s_1, s_2) (let us note that because of the connection in the switch : $a_2 = E_1$).

We have chosen a fixed value for the parameter of the input Bernoulli processes which is $p = 0.8$. This means that for any input port at each slot the probability that a cell arrives is 0.8. This value is chosen because it corresponds to a quite heavy traffic on the input ports of the switch.

To obtain the delayed geometrical output processes bounding the output stream of a queue at the first stage of the switch, we apply results on section 4 and section 5 to the Markov process denoted by $(., T^1)$ where T^1 is the transition matrix defined by (10). Applying results on section 4 we have :

$$\begin{pmatrix} b_0(p) & 1 - b_0(p) & 0 & \dots 0 \\ b_0(p) & 1 - b_0(p) & 0 & \dots 0 \\ \vdots & \vdots & \vdots & \vdots \\ b_0(p) & 1 - b_0(p) & 0 & \dots 0 \end{pmatrix} \leq_{st} T^1 \leq_{st} \begin{pmatrix} 0 & 0 & \dots 1 \\ 0 & 0 & \dots 1 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \dots 1 \end{pmatrix}$$

so we deduce that using aggregation technique $\rho_{inf}^{sl} = 1 - b_0(p)$ and $\rho_{sup}^{sl} = 1$ (this is the saturation case). An interpretation of the ρ_{inf}^{sl} can be done : the output process of a $Geom^X/D/FCFS/M_1$ is lower bounded by the output process of the particular queue $Geom^X/D/FCFS/1$.

For the configuration of the switch (2, 2, 2, 2) with $\rho_{sup}^{sl} = 1$ and $M_2 = 500$, the loss rate is only 7.810^{-4} but with $\rho_{sup}^{vc} = 0.956$ the loss rate value of 10^{-9} is reached as soon as M_2 is greater or equal to 100. So we see that the upper bound is more efficient using result of section 5 than the upper bound obtained using the result of section 4 (which is the worst case). That's why we focus our attention on bounds on the loss rate at the second stage obtained for geometrical parameter ρ_{inf}^{sl} and ρ_{sup}^{vc} .

In the table 1 we give the configuration of the switch, the values of the parameter of the bounding geometrical output stream ρ_{inf}^{sl} (recalling that this value is obtained using results of section 4) and ρ_{sup}^{vc} (recalling that this parameter is obtained using the algorithm of theorem 5.2), the capacity of a queue at the first stage such that loss rate is 10^{-9} , the capacities of queues at the second stage M_2^{sup} and M_2^{inf} such that the loss rate is equal to 10^{-9} when the arrival process at an input port of the second stage is geometrically distributed with parameter ρ_{sup}^{vc} and ρ_{inf}^{sl} , respectively.

The computation of the values in each column is done as follows. First we put a configuration of a switch (first column). Then we compute M_1 (fourth column) such that the loss rate is equal to 10^{-9} when input process is geometrically distributed with parameter $p = 0.8$. Then we compute ρ_{inf}^{sl} using results of section 4 and ρ_{sup}^{vc} using algorithm given in theorem 5.2. Then we finally compute M_2^{sup} and M_2^{inf} such that loss rate is equal to 10^{-9} by using the same procedure as for the computation of M_1 but with parameter p respectively equal to ρ_{sup}^{vc} and ρ_{inf}^{sl} .

(E_1, a_1, s_1, s_2)	ρ_{inf}^{sl}	ρ_{sup}^{vc}	M_1	M_2^{sup}	M_2^{inf}
(2, 2, 2, 2)	0.64	0.956	23	100	12
(10, 2, 2, 10)	0.64	0.956	23	172	21
(4, 2, 2, 8)	0.64	0.956	23	13	9
(4, 2, 4, 4)	0.36	0.617	7	17	9
(4, 2, 4, 8)	0.36	0.617	7	9	6
(5, 4, 4, 5)	0.59	0.968	34	210	17
(5, 4, 4, 10)	0.59	0.968	34	13	9
(2, 4, 8, 2)	0.344	0.672	10	15	6
(2, 4, 8, 4)	0.344	0.672	10	7	5
(2, 4, 8, 6)	0.344	0.672	10	5	4

Table 1

6.1 Discussion

First of all let us recall to the reader that the saturation case (i.e., when $\rho_{sup}^{sl} = 1$) is the worst case for some configurations as (2, 2, 2, 2) or (2, 4, 8, 2). But when the configuration is “good” (i.e., when the number of outputs is much greater than the number of inputs for all switching elements) the results indicate that this saturation case is efficient for addressing dimensioning problem. As an example for the configuration (2, 4, 8, 6) the loss rate value of 10^{-9} is obtained as soon as $M_2 > 6$ with ρ_{sup}^{sl} . The same value is obtained with ρ_{sup}^{vc} as soon as $M_2 > 4$.

For all configurations explored here the maximum error is about 1100% which could be sufficient to choose the best configurations which are here (4, 2, 4, 4), (4, 2, 4, 8), (2, 4, 8, 2), (2, 4, 8, 4) and (2, 4, 8, 6).

Last but not least, we have to mention here the most important result of this work. We have found approximated method which allows us to address dimensioning problem (of course for a simple model) and such that :

it computes an upper bound for the capacities of the queues at the second stage M_2^{sup} which insure that a queue with capacity M_2^{sup} is less than 10^{-9} ,

it computes a lower bound for the capacities of the queues at the second stage M_2^{inf} which allow us to compute an upper bound on the error made $M_2^{sup} - M_2^{inf}$.

As a final remark let us notice that this information is available for all possible configurations of the switch and we do not have to validate the results using rare event simulation.

7 CONCLUSION

We found bounding methodologies to address the dimensioning problem of a simple ATM switch. We found delayed geometrical bounds on the output stream of a $Geom^X/D/1/M$ queue with finite capacity M at the first stage of this switch model which allows to bound the loss rate. Except bounds based on the saturation case (which is the worst possible case) this method is only applied to answer dimensioning problem of queue of a switching element which is connected with independent queues from the previous stage. In other words we can imagine that this set of methods could be applied to address dimensioning problem of a Delta switch but we know at the first sight that this will give bad results when dimensioning the third stage of a Clos Network.

The key ideas of this work are to use results on lumpability and Veinott's Criterion which is a sufficient condition for the comparison of two random vectors in the sense of the strong ordering. In the two approaches the aim is to reduce the state space. For methods based on lumpability the idea is to find two stochastic matrices which are bounding a given stochastic matrix. The matrices found must be strongly lumpable. For method based on Veinott's Criterion we have noticed that in some cases it was exactly the same as the weak lumpability results (see [RS91] for instance). The Bounds we found are not optimal except when the transition matrix corresponding to the evolution of the number of customers in a $Geom^X/D/1/M$ is strong or weak lumpable. A case when this transition matrix is trivially strong lumpable is when the capacity M of the queue is equal to 1.

The complexity of the methods found are very interesting. Concerning methods based on the lumpability results (see section 4) their complexity for obtaining the loss rate at the second

stage is in $O(M_1^3 + M_2^3)$. Concerning the application of the Veinott's Criterion (see section 5) the computation of ρ_{sup}^{vc} is very fast (i.e., the computation time is very much lesser than the computation of the loss rate) in practice and the computation of the loss rate with ρ_{sup}^{vc} is still in $O(M_2^3)$.

What we want to stress is that the set of the methods presented in this paper contents methods which guarantee a lost rate less or equal to a given value (i.e, 10^{-9} in the ATM context) and give an upper bound on the error made. This result is obtained only using numerical methods without any simulation. The bounds could be improved and as an example let us notice that the upper bound obtained by saturation could easily be improved by leaving queues of the first stage on the input ports of a switching element of the second stage.

But these methodologies have to extended in two directions to having more importance in the ATM performance measuring community. The first one concerns the input traffics, the second one is the number of stages.

7.1 Input traffics

One of the most restrictive assumption made for this work concerns the input traffics wich are Bernoulli traffics. But if the input traffic is modeled by a Markovian process the results could again be applied. Of course the precision of the results will be worst. One of the possible track to avoid this is to modify the method based on Veinott's Criterion, but this is a further work.

7.2 Adding stages

One of the most problem is probably the problem of the performance measures of MIN with more than two stages. Assuming that the input processes are independent, the problem of the correlation input processes of switching element at the other stages is due moslty to the interconnection network of the switching elements. This means that the method proposed here could be adapted for a Delta network and the computation of the loss rate at the third stage. But the bounding methodologies could not be used to estimate the lost rate at the third stage of a Clos switch. Of course in a further work we have to focus our attention on this problem.

Acknowledgement : the author would like to thank Jean-Michel Couvreur from the Institut d'Informatique d'Entreprise (IIE-CNAM, France) for helpful discussion on the algebraic part of the technical proof of the theorem 5.1.

References

- [Bey93] A-L. Beylot. *Modèles de Trafics et de Commutateurs pour l'Evaluation de la Perte et du Délai dans les Réseaux ATM*. PhD thesis, Université de Paris 6, 1993.
- [BM93] A-L. Beylot and M.Becker. Performance Analysis of an ATM Clos Switching with Non Symmetric Switching Elements and Output Buffers. In *ITC sponsored seminar*

on Teletraffic for current and future Telecom Networks, Bangalore, India, November (15-19) 1993.

- [Clo53] C. Clos. A Study of Non-Blocking Switching Networks . Technical report, 1953.
- [Doi92] M. Doisy. *Comparaison de Processus Markoviens*. PhD thesis, Université de Pau et des Pays de l'Adour, 1992.
- [Gan64] F. R. Gantmacher. *The Theory of Matrices*, volume 2. Chelsea Pub. Company, 1964.
- [JJ60] Kemeny J.G and Snell J.L. *Finite Markov Chains*. Princeton, 1960.
- [Kei77] Adri Kester Julian Keilson. Monotone matrices and monotone Markov processes. *Stochastic processes and their applications*, 5, 1977. (231-241).
- [KHM87] M.J. Karol, M.G. Hluchyj, and S.P Morgan. Input vs Ouput Queuing on a Space Division Switch. *IEEE on Com.*, COM-3(12), December 1987.
- [Ros82] S.M. Ross. *Stochastic Processes*. J. Wiley & Son, 1982.
- [RS91] G. Rubino and B. Sericola. A finite characterization of weak lumpable Markov processes. Part I: The discrete time case. *Stochastic Processes and Their Applications*, 38, 1991. (195-204).
- [Sto76] D. Stoyan. *Comparaison Methods for Queues and other Stochastic Models*. J. Wiley and Son, 1976.

A Diffusion Cell Loss Estimate for ATM with Multiclass Bursty Traffic

Erol Gelenbe, Xiaowen Mang[†] and Yutao Feng
Department of Electrical and Computer Engineering
Box 90291, Duke University
Durham, NC 27708, USA
{erol,yf}@ee.duke.edu
Tel: (919) 660-5442
Fax: (919) 660-5293
[†]Cascade Communications Corp.
Westford, MA 01886, USA
xmang@ee.duke.edu or xmang@casc.com
Tel: (508) 952-1308
Fax: (508) 392-9250

Abstract

We describe a diffusion approximation model for an ATM statistical multiplexer using the instantaneous return model approach (Gelenbe, 1975). Two Cell Loss Estimates are proposed for multiclass traffic. Our aim is to provide a novel conservative, accurate and computationally efficient method for predicting cell loss probabilities which we call the *Finite Buffer Diffusion Cell Loss Estimate (FBDCLE)* and *Infinite Buffer Diffusion Cell Loss Estimate (IBDCLE)*. We evaluate their accuracy by comparing them with simulation results using a wide variety of input traffic characteristics. In particular we test the model with traffic which is a mixture of different “On-Off” sources with varying loads. Both homogeneous and heterogeneous aggregated arrival processes have been taken into account. These comparisons, which include evaluations of the statistical confidence of the simulation runs, show that our model predictions are very close to the simulation results. In particular, FBDCLE is a conservative upper bound to cell loss ratio, while the other (IBDCLE) provides an accurate predictor which may slightly under-estimate or over-estimate cell loss.

Keywords

ATM network performance prediction, quality of service, queueing theory, diffusion model, call admission control, bandwidth allocation.

1 INTRODUCTION

ATM provides a universal carrier service that can carry voice, data and video using the same cell transport arrangement. This technique allows complete flexibility in the choice of connection bit rate and enables the statistical multiplexing of variable bit rate traffic streams. On the other hand it also introduces a risk of overload, due to traffic variations which may cause network capacity to be exceeded. Overload is the main cause of cell loss and jitter in such systems. Thus the performance analysis of ATM multiplexers is critical to the design and analysis of appropriate control mechanisms for call admission, bandwidth allocation and bandwidth adaptation. Although much work has been done on the computation of cell loss ratios or probabilities which will result from a given ATM multiplexer in the presence of a given traffic (Kobayashi *et al.*, 1993) (Heffes *et al.*, 1986) (Sriram *et al.*, 1986) (Akimaru *et al.*, 1994), there is still much room for improvement in the methods used for finding computationally effective, fast and tight estimates of cell loss.

Typically, call admission and bandwidth adaptation controls use estimates of cell loss ratio for a given description of the incoming traffic at an ATM multiplexer or along a path traversing a series of multiplexers. For instance the call admission control policy used in IBM's ATM architectures (Guerin *et al.*, 1992) bases its bandwidth allocation conservatively using the minimum of two cell loss estimates: one based on equivalent bandwidth and the other on a Gaussian approximation of cell loss probability. Therefore more accurate estimates of cell loss probabilities will necessarily lead to better decisions for call admission. Thus it is important to be able to estimate cell loss ratios within a very wide range of variations ranging from 10^{-1} at the high end to less than 10^{-7} at the low end. It is important that the estimates obtained be conservative, i.e. that they be upper bounds, so that any bandwidth allocation based on these estimates does result in higher cell loss ratios. However, it is also essential that the estimate be a tight upper bound so that it will not result in the wasteful allocation of excessive bandwidth. Another consideration for any tool used for estimating cell loss is its computational cost. Many of the decisions making processes which use such estimates will have to be carried out in real time at low computational cost. Therefore our research aims at obtaining a tight, conservative and computationally effective method for estimating cell loss in an ATM multiplexer from given traffic characteristics. This paper uses diffusion approximations to contribute:

- a conservative cell loss ratio estimate we name FBDCLE (*Finite Buffer Diffusion Cell Loss Estimate*),
- and a tight estimate we call IBDCLE (*Infinite Buffer Diffusion Cell Loss Estimate*),

for superposed multiclass "On-Off" traffic. We use simulations to show the validity of FBDCLE and IBDCLE in the cell loss ratio range between 10^{-1} and 10^{-5} .

We describe the diffusion model in Section 2. In Section 3 and Section 4 we derive the FBDCLE and the IBDCLE. In section 5 we use the two estimates to compute cell loss ratios for multiple class "On-Off" traffic, and compare the analytical results with simulations for a wide variety of input traffic characteristics and different loads.

2 THE DIFFUSION MODEL

Diffusion approximations are continuous approximations to the discontinuous arrival and service processes in queueing models. They have long been used in queueing theory to model traffic and service. Their advantage is that they will generally result in computationally more tractable models of performance for more detailed traffic representations, that what can be obtained from a direct study of the corresponding discrete processes. In the past, two different approaches to diffusion approximations for queueing models have been proposed. In both cases whenever the queue length is non-zero and the maximum buffer capacity has not been attained, the queue length distribution is approximated by solving a partial differential equation. However the two methods differ according to the choice of boundary conditions. The simpler one uses reflecting boundaries (Kobayashi, 1974) (Kobayashi *et al.*, 1993) so that no probability mass accumulates at the boundaries. Clearly this approach will not be totally satisfactory if the boundaries themselves are very important to the process being modeled. The more sophisticated approach is based on the “instantaneous return process” (Gelenbe, 1975) (Gelenbe *et al.*, 1976) (Duda, 1986) which combines the partial differential equation formulation for the process *strictly* inside the boundaries, with a discrete state-space model at the boundaries themselves (Gelenbe, 1975). This leads to a more accurate model of the queueing behavior of the system when the load is low, or when the queue length is close to the maximum value allowed by a finite buffer.

Diffusion approximations require that the first two moments of the interarrival and service times be known. These can be directly deduced from measurements or from other traffic models, such as the “On-Off” model often used in the literature (Heffes *et al.*, 1986) (Sriram *et al.*, 1986). The diffusion approximation approach we take for an ATM multiplexer buffer of size B , considers a random process $\{X(t), t \geq 0\}$ to represent the buffer contents. In the open interval $]0, B[$ (excluding the two boundaries) it is a continuous random variable with probability density function $f(x, t)$ defined as:

$$f(x, t)dx = Pr[x \leq X(t) < x + dx], x \in]0, B[, \quad (1)$$

while at the boundaries we have:

$$m(t) = Pr[X(t) = 0], \quad (2)$$

$$M(t) = Pr[X(t) = B]. \quad (3)$$

The parameters for the diffusion process inside in $]0, B[$ are the “drift” or instantaneous average rate of change:

$$\mu = \lim_{\Delta t \rightarrow 0} \frac{E[X(t + \Delta t) - X(t) | X(t) \in]0, B[]}{\Delta t} \quad (4)$$

and the instantaneous variance of the change in $X(t)$:

$$\alpha = \lim_{\Delta t \rightarrow 0} \frac{Var[X(t + \Delta t) - X(t) | X(t) \in]0, B[]}{\Delta t} \quad (5)$$

and α will depend on the variance of the interarrival and service times at the ATM multiplexer. Since the service time is constant due to the fixed length of the cells being transmitted, α will only depend on the variance of interarrival times. Assuming time-independent traffic characteristics, let the mean aggregate cell arrival rate to the buffer be λ and the multiplexer cell transmission rate be C , both given in cells per second. Then we will have:

$$\mu = \lambda - C. \quad (6)$$

In the instantaneous return process model, when queue length reaches the lower boundary of the interval at $x = 0$, it remains there for a random length of random time which we denote h . This time clearly represents a period when the buffer is empty, and it ends as soon as a cell arrives to the multiplexer. At that time, say τ , the process $X(t)$ will jump from $X(\tau) = 0$ to $X(\tau^+) = +1$. Similarly for the upper boundary at $x = B$ where the random time spent at the boundary will be denoted by H , while the jump of the queue length process will be from the value B to the value $B - 1$ representing the end of a service or transmission epoch for a cell, resulting in a decrease of buffer length by 1. This behavior results in the following system of equations for the ATM multiplexer queue length process as derived in (Gelenbe, 1975) in steady state, where we have dropped the dependence on t :

$$-\mu \frac{\partial}{\partial x} f(x) + \frac{\alpha}{2} \frac{\partial^2}{\partial x^2} f(x) + \frac{m}{E[h]} \delta(x - 1) + \frac{M}{E[H]} \delta(x - B + 1) = 0 \quad (7)$$

$$\lim_{x \rightarrow 0^+} [-\mu f(x) + \frac{\alpha}{2} \frac{\partial f(x)}{\partial x}] = \frac{m}{E[h]} \lim_{x \rightarrow 0^+} \int f(x) dx = 0, \quad (8)$$

$$\lim_{x \rightarrow B^-} [-\mu f(x) + \frac{\alpha}{2} \frac{\partial f(x)}{\partial x}] = -\frac{M}{E[H]} \lim_{x \rightarrow B^-} \int f(x) dx = 0, \quad (9)$$

where $\delta(x)$ is the Dirac Delta function. Also the probabilities must sum to 1:

$$m + M + \int_{0^+}^{B^-} f(x) dx = 1 \quad (10)$$

These equations have a simple interpretation. Equation (7) represents the stationary behavior for the motion of the queue length process in the interval $]0, B[$, and the effect of the jumps of the process $X(t)$ from 0 and B into the interval. On the other hand (8) represents the depletion of the probability mass m at the lower boundary due to the jumps to $+1$ at the end of the holding time at the lower boundary, as well as the flow of probability mass from inside the interval $]0, B[$ towards the lower boundary. Equation (9) has a similar interpretation.

2.1 Queue length distribution of finite capacity

The above equations may be solved directly (Gelenbe, 1975) to obtain:

$$f(x) = \begin{cases} \Phi [1 - e^{\gamma x}], & 0 < x \leq 1 \\ \Phi [e^{-\gamma} - 1] e^{\gamma x}, & 1 \leq x \leq B - 1 \\ \Phi [e^{\gamma(x-B)} - 1] e^{\gamma(B-1)}, & B - 1 \leq x \leq B \end{cases} \quad (11)$$

with m and M the probability masses at 0 and at B , respectively, at stationary state being:

$$m = -\mu E[h] \Phi, \quad (12)$$

$$M = -\mu E[H] \Phi e^{\gamma(B-1)} \quad (13)$$

where $\gamma = \frac{2\mu}{\alpha}$, and

$$\Phi = \frac{1}{(1 - \mu E[h]) - (1 + \mu E[H]) e^{\gamma(B-1)}} \quad (14)$$

2.2 Queue length distribution of infinite capacity

If we consider a diffusion process on the whole non-negative real line, i.e. as if the queue length were infinite, with holding time h only at $x = 0$, we will have the following formula for an unbounded queue diffusion approximation model:

$$f(x) = \begin{cases} \Phi [1 - e^{\gamma x}], & 0 < x \leq 1 \\ \Phi [e^{-\gamma} - 1] e^{\gamma x}, & 1 \leq x \end{cases} \quad (15)$$

$$m = 1 - \Phi \quad (16)$$

$$\Phi = \frac{1}{(1 - \mu E[h])} \quad (17)$$

In the following sections, we will derive the practical applications of diffusion approximation models both for bounded queue and unbounded queue:

- Finite Buffer Diffusion Cell Loss Estimate (FBDCLC);
- Infinite Buffer Diffusion Cell Loss Estimate (IBDCLC).

In order to make use of these diffusion models we will need to determine the parameters μ , α , $E[h]$ and $E[H]$ from the arrival and service characteristics of the ATM multiplexer. From engineering application viewpoint of diffusion approximation models, various strategies can be used to obtain $E[h]$ and $E[H]$. More detail will be presented when we derive FBDCLC and IBDCLC.

3 FINITE BUFFER ESTIMATE - FBDCLE

In general the distributions for the residence times of moderately complex finite capacity queueing models at the upper and lower boundaries 0 and B are unknown. Their characterization can be quite complex and depends on both the arrival process, the buffer size, and the service process. Thus we will have to calculate $E[h]$ and $E[H]$ in a heuristic but plausible manner.

3.1 Calculation of $E[h]$ and $E[H]$

If the arrival process can be approximated by a Poisson process with arrival rate λ it follows that $E[h] = \lambda^{-1}$. Since the arrival traffic to an ATM multiplexer is made up of many superposed sources, when the number of sources is large this approximation may be acceptable. In our simulations it turns out that this heuristic for $E[h]$ slightly underestimates the actual value for superposed "On-Off" sources.

Recall that the time for transmitting one cell is C^{-1} . Now assume that at instant t the transmission of a cell begins and that $X(t) = B - 1$. At some instant $t + Z$ before $t + C^{-1}$ another arrival occurs so that now $X(t + Z) = B$. Then H , the random variable representing the holding time at the upper boundary, has the following distribution:

$$Pr[H \leq v] = Pr[\frac{1}{C} - Z \leq v | Z \leq \frac{1}{C}] = \frac{Pr[\frac{1}{C} - Z \leq v \text{ and } Z \leq \frac{1}{C}]}{Pr[Z \leq \frac{1}{C}]} \quad (18)$$

We make the simplifying approximation that the arrival process is Poisson of rate λ so as to complete the computation, on the basis that it is justified when the arriving traffic results from the superposition of many independent sources. Then

$$Pr[Z \leq \frac{1}{C}] = 1 - e^{-\frac{\lambda}{C}}, \quad (19)$$

and

$$Pr[\frac{1}{C} - Z \leq v \text{ and } Z \leq \frac{1}{C}] = Pr[\frac{1}{C} - v \leq Z \leq \frac{1}{C}] = e^{-\frac{\lambda}{C}} [e^{\lambda v} - 1]. \quad (20)$$

Thus

$$Pr[H \leq v] = \frac{e^{\lambda v} - 1}{e^{\frac{\lambda}{C}} - 1}, \quad (21)$$

with density function

$$f_H(v) = \begin{cases} \frac{\lambda e^{\lambda v}}{e^{\frac{\lambda}{C}} - 1}, & 0 \leq v \leq \frac{1}{C} \\ 0, & \text{elsewhere} \end{cases} \quad (22)$$

We can now derive the estimate for the average holding time at the upper boundary:

$$E[H] = \int_0^{\frac{1}{C}} v f_H(v) dv = \frac{\frac{1}{C}}{1 - e^{-\frac{\lambda}{C}}} - \frac{1}{\lambda}. \quad (23)$$

Of course, the first and second moments of the interarrival times are also needed in order to compute the density function $f(x)$ and the probability masses m and M . However, these moments will be available from the practical measurement and the precise traffic characteristics we shall use and will be discussed later in Section 5.

3.2 Estimating the cell loss ratio

The long run cell loss ratio L is the proportion of cells lost at the entrance to the multiplexer due to buffer overflow, to total cells arriving to the multiplexer. It is the primary measure of interest in this study and it needs to be estimated both accurately and in a conservative manner. Thus what is needed is in fact a tight upper bound, rather than a relatively accurate value which may underestimate L . Clearly cells will be lost only when the buffer is full, i.e. when buffer length has attained size B , in which case all the arriving cells will be lost. Thus the cell loss ratio in steady state may be written as:

$$L = \lim_{t \rightarrow \infty} M(t) Pr[N(t, t + H) \geq 1 \mid X(t) = B], \quad (24)$$

where $N(t, t + H)$ is the number of arrivals in the open interval $(t, t + H)$. If the arrival process is stationary in time and independent of buffer size, in steady state the expected cell loss ratio is:

$$L = M.Pr[N(t, t + H) \geq 1]. \quad (25)$$

There are several difficulties with using this expression when one deals with real traffic, including the issue of estimating H and the probability of the number of arrivals in the interval when the buffer is full. However we do know that $H \leq \frac{1}{C}$. Thus we have found that L_{FB}^* given below, which we call the *Finite Buffer Diffusion Cell Loss Estimate (FBDCLE)*, is a useful and tight upper bound which yields cell loss ratio values which are within the same order of magnitude as the value measured from simulation with various forms of "On-Off" traffic:

$$L \leq L_{FB}^* = M.Pr[N(t, t + \frac{1}{C}) \geq 1]. \quad (26)$$

The quality of this estimate L_{FB}^* has been tested by simulation with a very wide variety of "On-Off" traffic models, as shown in the simulation results we present.

4 INFINITE BUFFER ESTIMATE - IBDCLE

As indicated previously, the exact average residence times $E[h]$ and $E[H]$ of the finite capacity queueing model at the upper and lower boundaries are not known in general and are difficult

to obtain. Thus we consider an alternate formulation - infinite capacity queueing model where we only deal with the holding time at lower boundary $x = 0$. Now the key value used for estimating the cell loss probability will be the stationary probability that the diffusion process exceeds the value B :

$$P_B = Pr[X \geq B] \quad (27)$$

From (15) we estimate the buffer overflow probability P_B :

$$P_B = \Phi \frac{1}{\gamma} [1 - e^{-\gamma}] e^{\gamma B} \quad (28)$$

If $R(t)$ is the instantaneous cell arrival rate, then the new diffusion cell loss ratio estimate L is:

$$L = P_B \frac{E[(R(t) - C)^+]}{E[R(t)]} \quad (29)$$

since cell loss will only occur if the arrival rate is greater than the multiplexer's service capacity C whenever the buffer length is at least B .

4.1 Choice of $E[h]$

It is known that for the GI/GI/1 queue with arrival rate λ the average idle time $E[h]$ satisfies (Medhi, 1991):

$$E[h] \geq E[h]^* = \frac{1}{\lambda} - \frac{1}{C} \quad (30)$$

Thus we will approximate $E[h]$ by its lower bound $E[h]^*$, all other things being equal, the resulting probability P_B^* that the queue length exceeds B will be larger than real value P_B . This is because the process will be spending less time at $x = 0$ and therefore will be more likely to exceed B . This can also be easily proved by applying inequality of (30) into (28).

4.2 Estimating the cell loss ratio

The estimate L_{IB}^* , which we call the *Infinite Buffer Diffusion Cell Loss Estimate (IBDCLE)*, which in turn is obtained by replacing $E[h]$ by $E[h]^*$ in equation (28). IBDCLE will be:

$$L_{IB}^* = P_B^* \frac{E[(R(t) - C)^+]}{\lambda} \quad (31)$$

since $E[R(t)] = \lambda$ if $R(t)$ is stationary.

5 CELL LOSS ESTIMATES FOR “ON-OFF” MULTICLASS TRAFFIC

In this section we present the numerical and simulation results to evaluate the accuracy of FBDCLE and IBDCLE for a wide variety of “On-Off” traffic models. Much of the work on ATM traffic analysis and cell loss estimates is based on the “On-Off” traffic model and on the superposition of such traffic streams (Heffes *et al.*, 1986) (Sriram *et al.*, 1986). Thus it is of particular interest to evaluate the accuracy of our cell loss estimates (diffusion estimate) for this specific class of practically useful traffic models. In order to do so, we will first derive the appropriate traffic parameters to be used in the diffusion approximation.

5.1 The traffic model

Consider first a single user u whose traffic follows a simple “On-Off” behavior. This user u either sends traffic into the network at a constant peak rate R_u during the “On” period, or it sends no traffic at all during the “Off” period. The following notation describes this traffic model:

- R_u – peak traffic rate during the “On” period, $T_u = 1/R_u$;
- θ_u^{-1} – average length of the “Off” period;
- β_u^{-1} – average length of the “On” period;
- $a_u = \theta_u/(\beta_u + \theta_u)$ – source activity.

The duration of the successive On and Off periods are assumed to be independent, so that the cell arrival process from a single such source is a *renewal process*. The cell interarrival time will be denoted by Y_u , and let $F_u(x) = \Pr[Y_u \leq x]$ so that (Heffes *et al.*, 1986):

$$F_u(x) = [(1 - \beta_u T_u) + \beta_u T_u (1 - e^{-\theta_u(x - T_u)})] U(x - T_u) \quad (32)$$

where $U(x)$ is the unit step function. The Laplace-Stieltjes transform (LST) of the interarrival time density is given by:

$$\tilde{f}(s) = \int_0^\infty e^{-sx} dF_u(x) = [1 - \beta_u T_u + \beta_u T_u \theta_u / (s + \theta_u)] e^{-sT_u} \quad (33)$$

The mean cell arrival rate of cells from source u is then:

$$\lambda_u = -1/\tilde{f}'(0) = 1/(T_u + \beta_u T_u / \theta_u) = a_u / T_u = a_u R_u \quad (34)$$

Let $A_u(t)$ denote the number of arrivals of cells of user stream u in the interval $[0, t)$. Then the squared coefficient of variation of the interarrival time from source u is (Cox *et al.*, 1966) (Heffes *et al.*, 1986):

$$c_u^2 = \frac{\text{Var}[Y_u]}{E^2[Y_u]} = \frac{\text{Var}[A_u(t)]}{E[A_u(t)]} \quad (35)$$

which leads to (Heffes *et al.*, 1986):

$$c_u^2 = \frac{1 - (1 - \beta_u T_u)^2}{(\beta_u T_u + \theta_u T_u)^2}. \quad (36)$$

Since $E[A_u(t)] = \lambda_u t$, we can write (35) as:

$$\lim_{t \rightarrow \infty} \frac{Var[A_u(t)]}{t} = \lambda_u \frac{Var[Y_u]}{E^2[Y_u]} = \lambda_u c_u^2 \quad (37)$$

Now if the total arrival process to the ATM multiplexer results from the superposition of N uncorrelated "On-Off" sources of renewal type as discussed above, $A(t)$ the resulting counting process $A(t) = \sum_{u=1}^N A_u(t)$ has the obvious properties:

$$E[A(t)] = \sum_{u=1}^N E[A_u(t)], \quad (38)$$

$$Var[A(t)] = \sum_{u=1}^N Var[A_u(t)] \quad (39)$$

and

$$E[A(t)] = \sum_{u=1}^N \lambda_u t, \quad (40)$$

$$Var[A(t)] = \sum_{u=1}^N \lambda_u c_u^2 t \quad (41)$$

Let $D(t, t + \tau)$ denote the number of departures in an interval $[t, t + \tau]$ when the queue is non-empty. Note that if the multiplexer queue is non-empty, then the service or emptying process at the queue is independent of the arrival process. Thus we have:

$$E[X(t + \Delta t) - X(t) | X(t) > 0] = E[A(t + \Delta t) - A(t)] - E[D(t + \Delta t) - D(t)] \quad (42)$$

and

$$Var[X(t + \Delta t) - X(t) | X(t) \in]0, B[] = Var[A(t + \Delta t) - A(t)] + Var[D(t + \Delta t) - D(t)] \quad (43)$$

so that

$$\mu = \lim_{\Delta t \rightarrow 0} \frac{E[X(t + \Delta t) - X(t) | X(t) \in]0, B[]}{\Delta t} = \sum_{u=1}^N \lambda_u - C, \quad (44)$$

$$\alpha = \lim_{\Delta t \rightarrow 0} \frac{Var[X(t + \Delta t) - X(t) | X(t) \in]0, B[]}{\Delta t} = \sum_{u=1}^N \lambda_u c_u^2. \quad (45)$$

We now have all the parameters needed by the diffusion model described in Sections 2,3 and 4 when it is used for superposed “On-Off” traffic sources, and can use it to calculate the IBDCLE and FBDCLE formulae given in (26) and (31).

5.2 The distribution of the number of arrivals

In order to calculate the FBDCLE, the quantity $Pr[N(t, t + \frac{1}{C})]$ must be obtained. To do so, we will consider the general case of arrival traffic composed of multiple “On-Off” sources of K different *types*. Each source of the same type will have the same set of parameters, and N_k will be the number of k -type sources, each with the same peak traffic rate R_k , activity a_k . Notice that here we use the subscript k to denote a user type, rather than the subscript u to denote an individual user. The total number of users or sources is then $N = \sum_{k=1}^K N_k$. The average arrival rate of cells will then be:

$$\lambda = \sum_{k=1}^K a_k N_k R_k \quad (46)$$

Now let $Z_k(t)$ be the random variable denoting the number of sources of type k which are “On” at some time t . Since the sources are independent and stationary we have for large enough t that:

$$Pr[Z_1(t) = n_1, \dots, Z_K(t) = n_K] = \prod_{k=1}^K \binom{N_k}{n_k} a_k^{n_k} (1 - a_k)^{N_k - n_k} \quad (47)$$

On the other hand for small enough $1/C$:

$$N(t, t + \frac{1}{C}) = [Z_1(t)R_1 + \dots + Z_K(t)R_K]/C, \quad (48)$$

so that:

$$Pr[N(t, t + \frac{1}{C}) \geq 1] = Pr[Z_1(t)R_1 + \dots + Z_K(t)R_K \geq C], \quad (49)$$

which can be computed from the distribution (47). For homogeneous traffic, i.e. when all sources are of just one type, we simply have $K = 1$ and:

$$Pr[N(t, t + \frac{1}{C}) \geq 1] = 1 - \sum_{n_1=1}^{int(C/R_1)} \binom{N_1}{n_1} a_1^{n_1} (1 - a_1)^{N_1 - n_1}. \quad (50)$$

For the IBDCLE we need $E[(R(t) - C)^+]$ to be used in (31), which is computed for the superposed multiclass “On-Off” traffic as:

$$E[(R(t) - C)^+] = \sum_{n_1, \dots, n_K \geq 0} (n_1 R_1 + \dots + n_K R_K - C)^+ Pr[Z_1(t) = n_1, \dots, Z_K(t) = n_K] \quad (51)$$

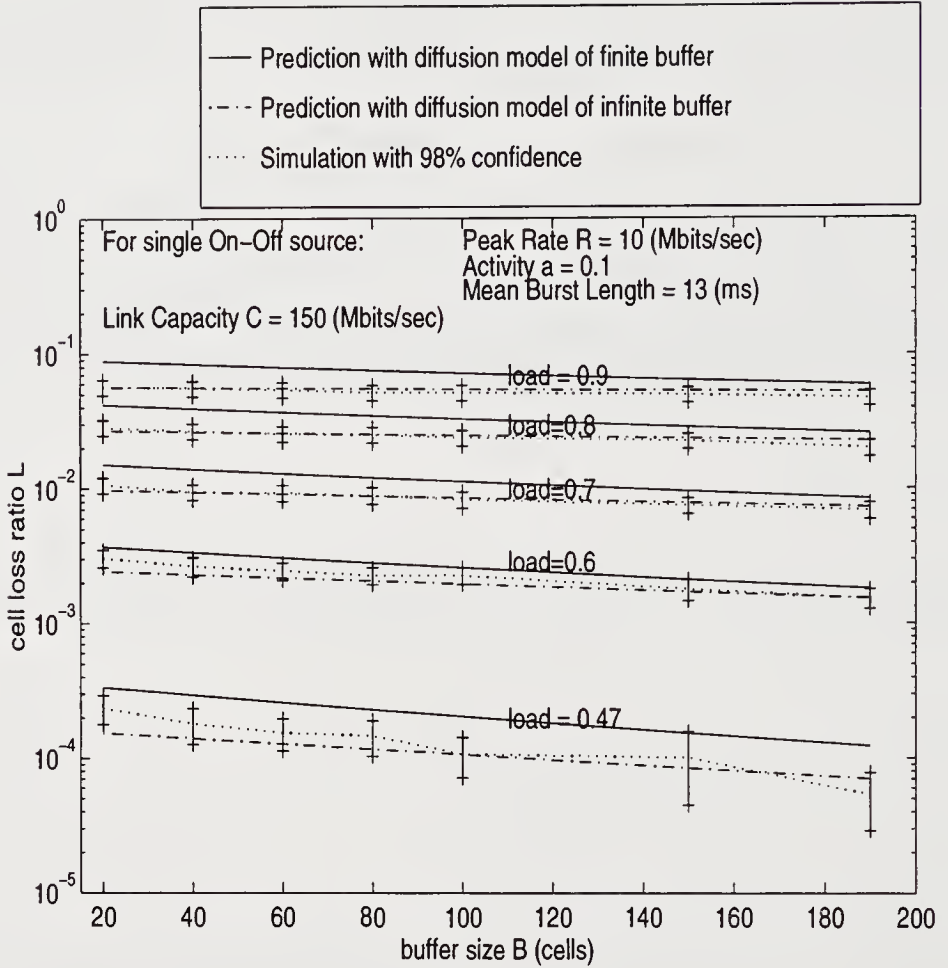


Figure 1 Cell loss probability vs. buffer size: comparison of simulation and DCLE for homogeneous sources under varying load (load = aggregate mean arrival rate /link capacity).

5.3 Comparison of numerical and simulation Results

In this section we present the numerical and simulation results to evaluate the accuracy of FBDCLC and IBDCLE. The validation of our new diffusion model is focused on the comparison of the cell loss probability predicted by the FBDCLC and IBDCLE and that obtained by simulations for a wide variety of “On-Off” traffic models. In our simulations, the runs were independently replicated 20 times, and each run included the transmission of 10^7 cells. Confidence intervals are calculated using the *Student - t* distribution with 98% confidence so that the simulation results are of sufficiently high statistical quality. The resulting confidence intervals’ width is also shown on the figures.

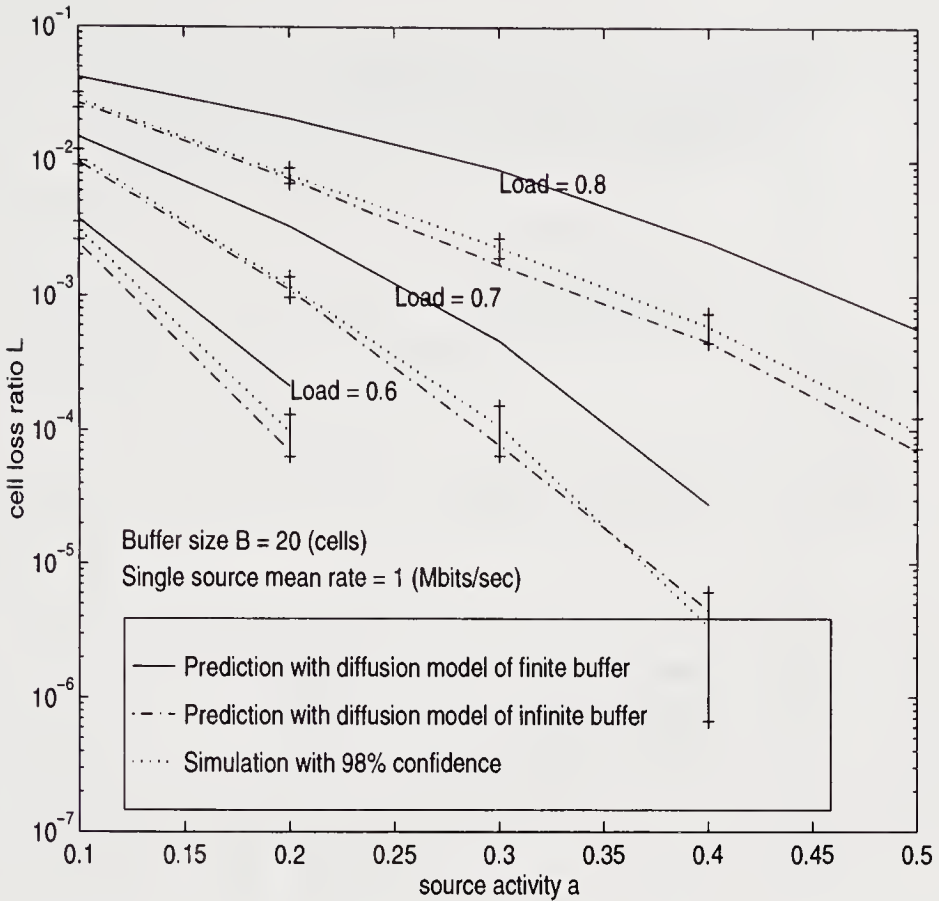


Figure 2 Cell loss probability vs. source activity (burstiness): comparison among simulations and analytical approach using DCLE for the homogeneous sources under variant load (load = aggregate mean rate /link capacity).

Figures 1 and 2 summarize the results for traffic with homogeneous sources.

In Figure 1 cell loss probability ($Pr[\text{cell loss}]$) is plotted versus buffer size B for different load, which is λ/C . The ATM multiplexer we consider here is a high speed link with link capacity $C = 150 \text{ Mbits/sec}$ and there are a collection of homogeneous traffic sources which are very bursty with an activity value of $a = 0.1$, which means that it is at its peak value 10% of the time and is "Off" the rest of the time. Load is varied in Figure 1 simply by varying the number of sources. The results show that for cell loss ratio ranging from the high 10^{-5} to the 10^{-1} values, the FBDCLE (the solid line) provides a conservative upper bound, while the IBDCLE (the dashed and dotted line) is an accurate predictor which remains well

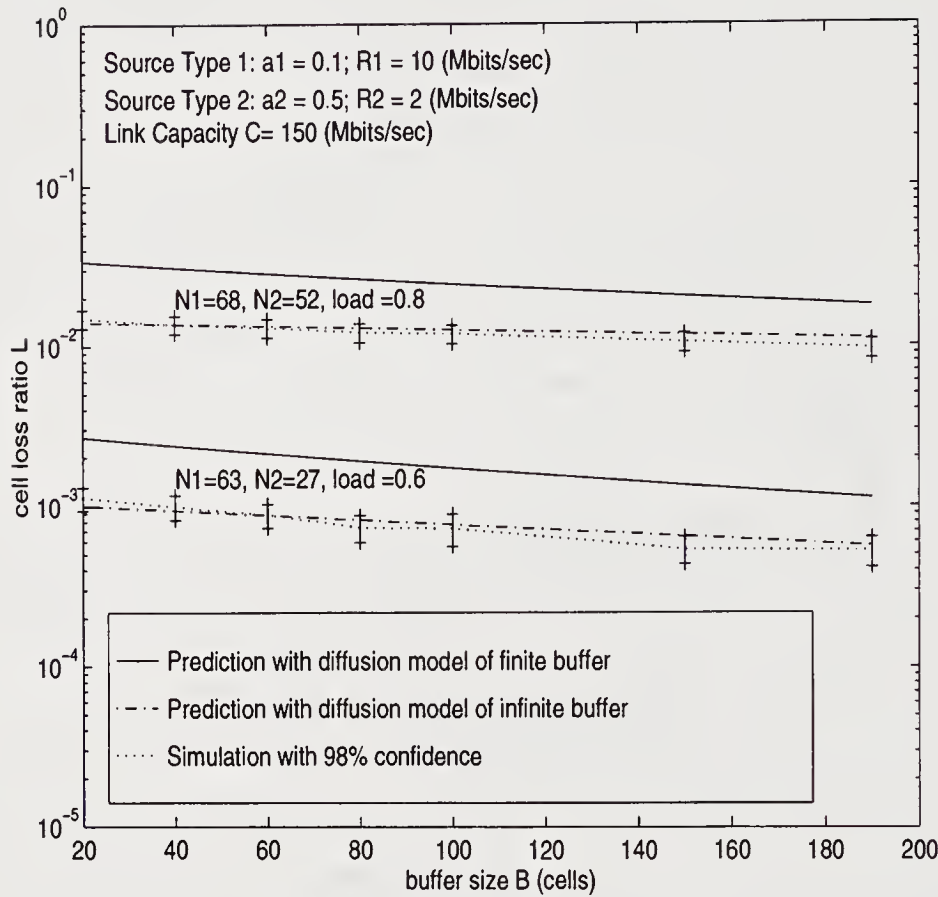


Figure 3 Cell loss probability versus buffer size: comparison between simulation and DCLE for heterogeneous sources with varying load (load = aggregate mean rate /link capacity).

within the confidence intervals. Simulation results are shown by the dotted lines while the 98% confidence intervals are vertical lines.

In Figure 2 similar results are observed when source activity a (or burstiness) is varied widely for different values of the load. Here each individual source generates cells at an average rate $\lambda_u = 1$ (Mbits/sec) and the buffer size is relatively small: $B = 20$ cells. Note that here we see that IBDCLE is an accurate predictor over cell loss ratio values ranging from 5×10^{-6} to 3×10^{-2} .

Figures 3 and 4 compare FBDCLE and IBDCLE with simulation under heterogeneous traffic. We have chosen two types of sources – more bursty sources with $a_u = 0.1$ and less

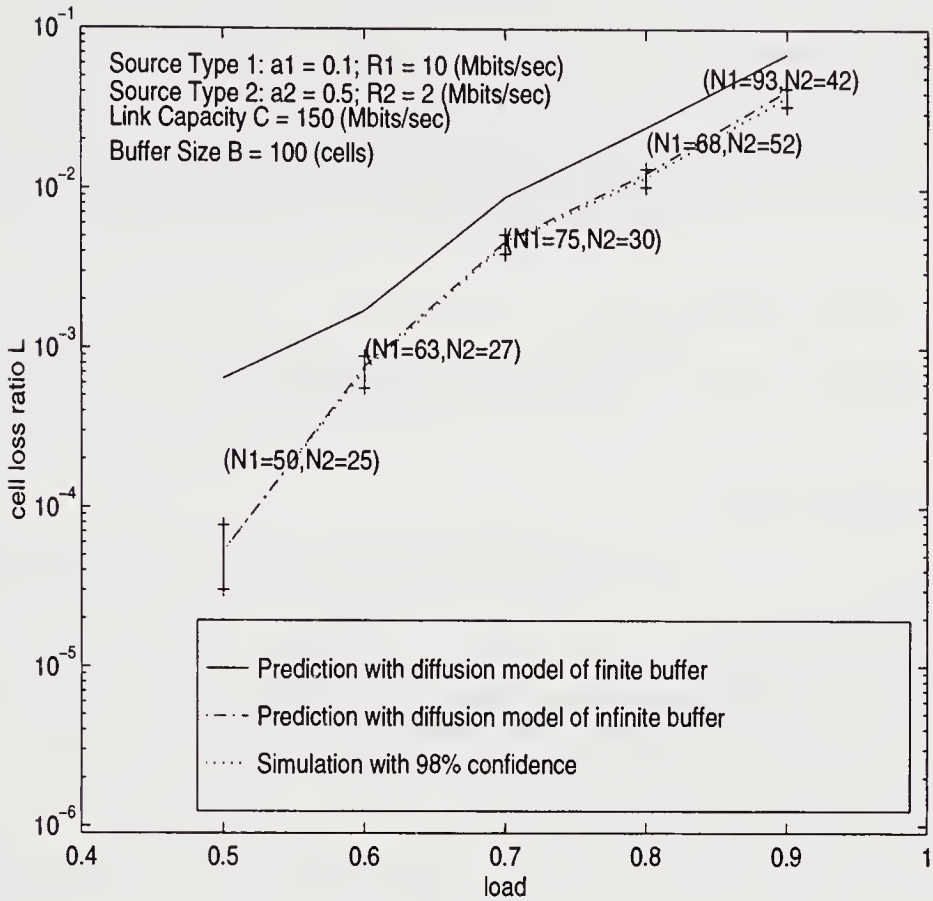


Figure 4 Cell loss probability versus load: comparison between simulation and the DCLE for heterogeneous sources (load = aggregate mean rate / link capacity).

bursty sources with $a_u = 0.5$. If N_1 and N_2 denote the number of sources with $a_u = 0.1$ and $a_u = 0.5$ respectively, and $N = N_1 + N_2$.

In Figure 3 we show matched results of simulations and the diffusion predictions for two different values of the load, and under different combinations of N_1 and N_2 with varying buffer size B . Note that the two classes are also characterized by two much different values of peak traffic rate: $R_1 = 10$ (Mbits/sec) and $R_2 = 2$ (Mbits/sec). Again we see that the FBDCLE (the solid line) gives a bounded estimate while IBDCLE provides a very accurate prediction (the dashed and dotted line).

In Figure 4 the cell loss probability is plotted versus traffic load for a fixed buffer size $B = 100$, the same two-class traffic as in Figure 3 and five different load values obtained by

varying the mixture of class 1 and class 2 traffic. The simulation results, together with their confidence intervals, show once again excellent agreement with our infinite buffer estimate (IBDCLE) while the FBDCLE is again an upper bound, for cell loss ratio values going from 5×10^{-5} to 3×10^{-2} .

We conclude from these results, and from others which are available but which are not reported here because of space limitation, that the FBDCLE can be used for a very conservative estimate of cell loss, while IBDCLE is useful as an accurate predictor.

6 ACKNOWLEDGMENT

The authors are grateful to the support from IBM Corporation. However this paper only represents the views of the authors.

7 REFERENCES

- D. R. Cox and P. A. W. Lewis (1966). *The Statistical Analysis of Series of Events*. Methuen, London.
- A. Duda (1986). Diffusion approximations for time-dependent queueing systems. *IEEE J. SAC*, SAC-4(6), 905-18.
- E. Gelenbe (1975). On approximate computer system models. *J. ACM*, **22**, 261-3.
- R. Guerin and L. Gun (1992). A unified approach to bandwidth allocation and access control in fast packet-switched networks. *Proc. INFOCOM'92*, 1-12.
- E. Gelenbe and G. Pujolle (1976) An approximation to the behaviour of a single queue in a network. *Acta Informatica*, **7**, 123-36.
- H. Akimaru, T. Okuda and K. Nagai (1994) A Simplified Performance Evaluation for Bursty Multiclass Traffic in ATM Systems. *IEEE Transactions on Communications*, COM-42(5), 2078-83.
- H. Heffes and D. M. Lucantoni (1986) A markov modulated characterization of packetized voice and data traffic and related statistical multiplexer performance. *IEEE J. SAC*, SAC-4(6), 856-67.
- H. Kobayashi (1974) Application of the diffusion approximation to queueing networks: Parts I and II. *J. ACM*, **21**, 316-28.
- H. Kobayashi and Q. Ren (1993) A diffusion approximation analysis of an ATM statistical multiplexer with multiple state solutions: Part I: Equilibrium state solutions. *Proc. ICC'93*, 1047-53.
- J. Medhi (1991) *Stochastic Models in Queueing Theory*. Academic Press, New York.
- K. Sriram and W. Whitt (1986) Characterizing superposition arrival processes in packet multiplexers for voice and data. *IEEE J. SAC*, SAC-4(6), 833-46.

An integrated approach to evaluating the loss performance of ATM switches

S. Montagna

Italtel - A STET and Siemens Company, Milan, Italy

tel. 39 2 4388 8098, fax. 39 2 4388 7989

e.mail Montagna@settimo.italtel.it

R. Paglino

Italtel - A STET and Siemens Company, Milan, Italy

J. F. Meyer

*Department of Electrical Engineering and Computer Science,
The University of Michigan, Ann Arbor, Michigan, USA.*

Abstract

This work addresses model-based evaluation of cell loss probabilities for an ATM switching element with a shared output buffer. The incoming traffic to the switch is represented by the superposition of N bursty input sources, each of which is modeled as a two-state (On/Off) Markov chain. For such systems, we consider an integrated approach to their evaluation that employs both exact and approximate solutions. The exact method is based on a reduced Markov model obtained by lumping the states according to certain symmetries of the traffic model. However, even with such reduction, numerical solutions are feasible only if the switch dimensions involved, particularly the number of output ports, are reasonably small. We then introduce a new approximate solution algorithm that can be applied to larger switches. By comparing the results obtained with those of the exact method, we find that the errors of approximation are relatively small. Moreover, due to the iterative nature of the approximate solution algorithm, the two methods can be integrated so as to yield even more accurate results with less execution time.

Keywords

ATM switch, shared buffers, On/Off sources

1 INTRODUCTION

Services both realized and planned for broadband, ATM-based, ISDNs impose extremely severe constraints on the performance of ATM switching elements. In particular, admissible cell loss probabilities as small as 10^{-9} (or even less) call for switch buffers that are sufficiently large to guarantee this quality of service. In this regard, it has been shown (see [1,2], for example) that the best utilization of buffer capacity is obtained by dynamically sharing cell storage among all the output ports of the switch. This permits a reduction of required capacity (for a specified admissible cell loss probability) relative to switches which employ dedicated, fixed-capacity queues at either the input or the output. However, the problem of evaluating the loss performance of a shared-buffer switch is difficult, due primarily to the large number of internal states that must be accounted for in the process, even when the switch dimensions are relatively small. Therefore, various studies have proposed approximate solutions to this problem, assuming further (see [3,4,5], for example) that traffic sources for the input ports are represented by independent Bernoulli arrival processes, thus precluding any correlation between cell arrivals. Among such investigations, perhaps the most widely cited is [3] which presumes an infinite buffer and approximates its steady-state occupancy distribution with a Gamma function. The parameters of the Gamma distribution are obtained analytically by computing the mean and variance of the shared-buffer occupancy distribution. This method provides a practical means of quickly estimating the required buffer capacity of a switch. However, since it matches only the first two moments of the actual distribution, it often fails to accurately estimate the distribution's "tail", i.e., the probabilities of large occupancies which have very low values.

Another simple way to estimate the buffer occupancy distribution of a shared buffer with uncorrelated traffic is by convolving the individual distributions of a number of Geo/D/1 queues. Since Geo/D/1 models are relatively easy to solve (as discussed in [6], for example), this method is also attractive. Other studies, such as those of [7,4], suggest more complex heuristic algorithms that typically lead to more precise solutions.

Although Bernoulli sources are convenient by virtue of their simplicity, a more realistic arrival process should capture correlation that exists between successive arrivals at an input port. This is done in [8], for example, by employing a continuous-time model where each input source is modeled by an interrupted Poisson process. The investigation that follows considers two discrete-time models of a shared-buffer switch subjected to bursty (and hence correlated) traffic. They support exact and approximate solution algorithms, respectively; moreover, we find that these methods can be usefully integrated to achieve both greater accuracy and reduced execution time (when compared with exclusive use of the approximate method).

The first method, referred to as *Algorithm 1*, provides an exact solution of the steady-state distribution of shared-buffer occupancy for switches of limited (but not trivial) size. This solution is based on an efficient representation of the state space that derives from certain symmetries implied by the underlying assumptions. Although its application is limited in the sense noted above (its execution time grows exponentially with buffer capacity), it is nevertheless very useful. In particular, in addition to providing exact results for the probabilities of rare cell-loss events, it can serve as a reference for assessing the nature and

magnitude of errors that result from approximate analytic models and/or solution techniques. Although simulation is often used for this purpose, such practice is reasonable only if the simulation results are themselves highly accurate (high confidence with respect to small confidence intervals).

Further, as we emphasize in the development that follows, it is sometimes possible to integrate the use of exact solutions with certain types of approximation techniques, thereby extending the scope of the former. For example, if the approximation algorithm is iterative in nature (as in the case of convolution, or more specifically, the algorithm we consider below) then an exact solution can be usefully employed for the initial iteration. This leads to more accurate approximate evaluations, even for realistically large switches with bursty traffic.

The approximation technique we propose is new (*Algorithm 2*) and is based on a decomposition of the system into smaller systems involving fewer output ports. Comparisons (see section 4) of algorithm-2 results with those of algorithm 1 (for small switch sizes and very low loss requirements) and with simulation data (for larger systems with relatively high losses) reveal that the approximations obtained are reasonably accurate. These results are then used to estimate the advantage, in terms of memory saving, of a shared-buffer architecture relative to a simpler architecture that employs a dedicated, fixed-capacity buffer for each output port. With such estimations, the required shared-buffer capacities for very low admissible cell loss probabilities can be likewise estimated.

Assumptions concerning the switch and its traffic are discussed in section 2. This is followed by descriptions of the two algorithms, including their integration (section 3) and, in turn, a presentation of the results just mentioned (section 4). Section 5 then summarizes what was accomplished, with appendices A and B providing some solution details that were omitted in section 3.

2 THE SWITCH

The switch considered has a typical shared-buffer architecture, i.e., memory space available to store ATM cells is dynamically shared among all the output queues. Incoming cells arrive from N input ports and are addressed to one of R output ports. Provided there is available space in a common buffer of finite capacity K (the maximum number of cells that can be stored), via an appropriate pointer structure (maintained in a separate memory space), a cell is then stored in a logical FIFO output queue corresponding to its address. A cell is lost if and only if no buffer space is available when the cell arrives. The switch is assumed to operate synchronously at the cell level; in a given time slot (the time required to completely transmit/receive a cell on a port of the switch), we presume that the following two operations take place in the order indicated.

Send: For each non-empty logical queue, the least recently arrived cell is served and its buffer space is freed.

Receive: Each incoming cell is stored in the buffer (if there is available space) and the pointer chain is appropriately updated; these cells will be served in the next slot.

The traffic at each of the N input ports is represented by a 2-state (On/Off) Markov chain where these individual sources are assumed to be statistically independent. In the On state, a

cell arrives with probability 1 while in the Off state there are no arrivals. The dwell times in each state (number of time slots between entry and exit) are geometrically distributed variables, with means L and I for the On and the Off states, respectively.

The activity ρ_{in} of an individual source is the fraction of time the source is in the On state and is given by $\rho_{in} = L/(I+L)$. The destination address of each cell (i.e. the output port it is queued to) is a random variable that's uniformly distributed over the R output ports and is independent of the destinations of previously arrived cells. This assumption attempts to capture the situation where each input link carries the superposition of a large number of low bit-rate connections, each connection addressed to a possibly different output link.

As is well known, a purely random (memoryless) traffic model, where each input behaves as a Bernoulli source, is a special case of the model just described. Specifically, the above reduces to the Bernoulli case if $L = 1/(1 - \rho_{in})$ and $I = 1/\rho_{in}$. Finally, we let ρ denote the offered load, as reflected by the utilization of an output port (assuming no cell losses), i.e., $\rho = N \cdot \rho_{in} / R$.

3 THE ALGORITHMS

As mentioned in our introductory remarks, we choose to employ both exact and approximate model-based methods to determine the steady-state probability distribution of shared-buffer occupancy, given the switch/traffic assumptions stated above. (Other measures, such as loss probability are then based on this distribution.) These are described in the subsections that follow, with some of the mathematical details being deferred to appendix A (algorithm 1) and appendix B (algorithm 2). However, before proceeding with these descriptions, it is helpful to introduce some assumptions, terminology, and notation which are common to both algorithms.

Time is assumed to be discrete, where a time instant t takes values in the set $T = \{0, 1, 2, \dots\}$. The duration between successive instants t and $t + 1$ is interpreted as the t th time slot, where the enumeration begins with time slot 0 and instant t represents the beginning of slot t , i.e., it occurs before the intraslot "send" and "receive" operations described in section 2.

M_t = number of sources in the On state during slot t

$X_{i,t}$ = number of cells in the buffer at time t addressed to output port i , $i = 1, 2, \dots, R$

$Y_t = \sum_{i=1}^R X_{i,t}$ = number of cells in the buffer at time t .

By these definitions, along with our earlier assumptions concerning switch dimensions N and K , for any $t \in T$, these variables are thus constrained to have integer values in the ranges

$$0 \leq M_t \leq N \tag{1}$$

$$0 \leq X_{i,t} \leq K; \quad i = 1, 2, \dots, R \tag{2}$$

$$0 \leq Y_t \leq K \tag{3}$$

Since there is an arrival at an input port if and only if the source for that port is in the On state, M_t is just the number of arrivals during slot t , including some that may be lost if the

buffer is full. However, if the buffer capacity is at least N (which we tacitly assume throughout the discussion) then, for all $t \in T$,

$$M_t \leq Y_t \quad (4)$$

3.1 Algorithm 1 (Exact)

Let $X_t = (X_{1,t}, X_{2,t}, \dots, X_{R,t})$ be the R -dimensional vector-valued random variable that represents the cell-occupancy of the shared buffer at time t . If, further, we let X denote the corresponding stochastic process, i.e., $X = \{X_t | t \in T\}$ then, without simplification, the state space Q of X quickly becomes computationally intractable, even for relatively small values of R and K . For example, if $R = 8$ and $K = 40$ then $|Q| \approx 4 \cdot 10^8$. ($|Q|$ is the cardinality of the state space Q .)

A key observation that drastically reduces the size of the state space (while still supporting an exact solution) is the following. Due to assumptions concerning i) the identical probabilistic nature of individual input sources and ii) the uniformity of cell routing, it is possible to lump (partition) the state space Q according to the following equivalence relation. Letting q_i denote the number of cells in the shared buffer that are destined for output port i ($i = 1, 2, \dots, R$), two states $q = (q_1, q_2, \dots, q_R)$ and $q' = (q'_1, q'_2, \dots, q'_R)$ are *equivalent* (and, hence, in the same lump) if and only if q' is a permutation of q . Letting \bar{Q} denote the resulting partition of Q , it then suffices to consider the corresponding reduced stochastic process $\bar{X} = \{\bar{X}_t | t \in T\}$ where, for all $t \in T$, \bar{X}_t is the equivalence class (lump) that contains state X_t . For various choices of queue capacity K , the extent of this reduction is indicated in Tables 1 and 2, where the number of output ports is $R = 4$ and $R = 8$, respectively. Specifically, these tables compare the size of the original state space Q with that of the reduced space \bar{Q} , where we see that reductions of several orders of magnitude are possible.

Table 1: State-space size reduction if $R = 4$.

	$K = 10$	$K = 20$	$K = 40$	$K = 80$
$ Q $	10^3	10^4	$1.3 \cdot 10^5$	$1.9 \cdot 10^6$
$ \bar{Q} $	94	$7.1 \cdot 10^2$	$7.3 \cdot 10^3$	$9.2 \cdot 10^4$

Table 2: State-space size reduction if $R = 8$.

	$K = 10$	$K = 20$	$K = 40$	$K = 80$
$ Q $	$4.3 \cdot 10^3$	$3.1 \cdot 10^6$	$3.8 \cdot 10^8$	$7.4 \cdot 10^9$
$ \bar{Q} $	$1.3 \cdot 10^2$	$2.0 \cdot 10^3$	$7.3 \cdot 10^4$	$8.1 \cdot 10^5$

To obtain a feasible means of determining the steady-state probability distribution of \bar{X} , each state $\bar{q} \in \bar{Q}$ can be conveniently represented by an ordered pair (b, e) consisting of a

sequence of “occupancy values” b and an “occupancy vector” e (see appendix A). Using this representation, algorithm 1 is based on a functional formulation of transitions to intermediate states (during a slot) that result from the “send” operation and, in turn, each intraslot arrival. Beginning with state \bar{X}_t , which expresses the buffer occupancy at the end of slot t , and accounting for these intraslot transitions during slot $t+1$, the resulting state (following the final cell arrival) is then the next state \bar{X}_{t+1} of the lumped buffer model. (Again, see appendix A for further details.) Accordingly, if we account for the behavior of the Markovian source model $M = \{ M_t | t \in T \}$ then, given that $\bar{X}_t = (b, e)$ and $M_t = m$ (the number of On sources during slot t), these functions, together with the transition probabilities of M , determine the conditional probabilities

$$P[\bar{X}_{t+1} = (b', e'), M_{t+1} = m' | \bar{X}_t = (b, e), M_t = m] \quad (5)$$

for all $(b', e') \in \bar{Q}$ and all $m' \in \{0, 1, \dots, N\}$. In other words, if we let Z_t be the pair of variables (\bar{X}_t, M_t) and consider the corresponding stochastic process $Z = \{ Z_t | t \in T \}$ then the transition probabilities of Z at time t are given by (5). Beginning with some arbitrary distribution for the initial state variable $Z_0 = (\bar{X}_0, M_0)$ the distribution of Z_{t+1} can thus be determined iteratively from the distribution of Z_t and the transition probabilities (5) at time t , for $t = 0, 1, 2, \dots$ until a steady-state (stationary) condition is sufficiently well approximated. More precisely, the computation terminates when, for all $(b, e) \in \bar{Q}$ and all $m \in \{0, 1, \dots, N\}$ the absolute value of the relative difference

$$\frac{P[Z_{t+1} = ((b, e), m)] - P[Z_t = ((b, e), m)]}{P[Z_t = ((b, e), m)]}$$

is less than some very small positive number. Given that t satisfies this condition, the distribution we seek is then obtained by summing over the states of the source model M , i.e., for all $(b, e) \in \bar{Q}$,

$$P[\bar{X}_t = (b, e)] = \sum_{m=0}^N P[Z_t = ((b, e), m)].$$

Although application of this algorithm becomes intractable for large values of R and K , as indicated in tables 1 and 2, the reduction in state space size provided by the reduced model permits feasible solutions for switches with moderate dimensions. For example, when implemented on an HP9000 series 700 workstation, algorithm 1 can accommodate an 8×8 switch with random traffic and a buffer capacity of $K=60$ or bursty traffic and a buffer capacity of $K=40$. Moreover, and as noted in section 1, exact models are likewise very useful if a large system can be approximately decomposed into smaller subsystems that admit to such representation. For instance, in addition to more obvious uses such as convolution, this type of “divide and conquer” approach was employed by the approximate method developed

in [3]. Details as to how algorithm 1 can be exploited in concert with algorithm 2 are presented at the end of the subsection that follows.

3.2 Algorithm 2 (Approximate)

This algorithm approximates the shared-buffer occupancy probabilities via an iterative procedure that considers, at each successive step, subsystems of growing size. Presuming an $N \times R$ switch with a shared buffer of capacity K , for a specified integer r , where $1 \leq r \leq R/2$, we initially choose two disjoint subsets $1(r)$ and $2(r)$ of the set $\{1, 2, \dots, R\}$ of all output ports, where each subset has cardinality r . Although just how these subsets are chosen is relatively arbitrary, to simplify the discussion we assume that both R and r are integer powers of 2. Moreover, without loss of generality, we can let $1(r)$ be output ports 1 through r and $2(r)$ be ports $r+1$ through $2r$, i.e.,

$$1(r) = \{1, 2, \dots, r\} \text{ and } 2(r) = \{r+1, r+2, \dots, 2r\}.$$

The shared buffer, together with the $2r$ output ports $1(r) \cup 2(r)$, will be referred to simply as an r -subsystem. In keeping with the notation of the exact method, the buffer state at a given time t is described by the random variables

$$X_{i(r),t} = \text{number of cells in the buffer at time } t \text{ addressed to output ports in } i(r), i=1,2$$

and, to represent arrivals and departures in an analogous fashion, we let

$$W_{i(r),t} = \text{number of incoming cells at time } t \text{ addressed to ports in } i(r), i=1,2$$

$$Z_{i(r),t} = \text{number of cells that depart the buffer at time } t \text{ from output ports in } i(r), i=1,2.$$

Relative to this model of an r -subsystem, and recalling that M_t is the number of sources in the On state at time t , let us now consider the following limiting distributions concerning arrivals (A_r), combined buffer occupancy and source activity (B_r , referred to as the "buffer-source" distribution), and departures (D_r).

$$A_r(w_1, w_2 | m) = \lim_{t \rightarrow \infty} P[W_{1(r),t} = w_1, W_{2(r),t} = w_2 | M_t = m] \quad (6)$$

$$B_r(x_1, x_2, m) = \lim_{t \rightarrow \infty} P[X_{1(r),t} = x_1, X_{2(r),t} = x_2, M_t = m] \quad (7)$$

$$\begin{aligned} D_r(z | x, m) &= \lim_{t \rightarrow \infty} P[Z_{1(r),t} = z | X_{1(r),t} = x, M_t = m] \\ &= \lim_{t \rightarrow \infty} P[Z_{2(r),t} = z | X_{2(r),t} = x, M_t = m] \end{aligned} \quad (8)$$

Computation of the conditional arrival probabilities $A_r(w_1, w_2 | m)$ is straightforward since, by the uniform routing assumption, each arriving cell has a probability $1/R$ of being addressed to a given output port. Hence, as both $1(r)$ and $2(r)$ have cardinality r , for either subset $i(r)$ ($i=1,2$), the probability of an arrival being addressed to a port in $i(r)$ is simply r/R . With this observation, let $BI_{n,p}$ denote the binomial distribution having parameters n and p , i.e., for $0 \leq i \leq n$,

$$BI_{n,p}(i) = \binom{n}{i} p^i (1-p)^{n-i}.$$

Then in case all arrivals are accepted (the “no-loss” case), the formulation of A_r is immediate, i.e.,

$$A_r(w_1, w_2 | m) = BI_{m, 2r/R}(w_1 + w_2) \cdot BI_{w_1 + w_2, 1/2}(w_1). \quad (9)$$

To extend (9) so that it can account for cell losses, we assume further that there is no statistical dependence between the address of an arriving cell and the event that it is one the cells discarded among the m that arrive. In this case, the extension is easily obtained.

The departure distribution D_r , on the other hand, is more difficult to determine once the value of r is greater than 1. It is here that we choose to introduce an approximate computation based on the following recursive formulation of D_r in terms of $D_{r/2}$ and $B_{r/2}$ (where $r > 1$). (The inexact nature of this formula will be discussed in a moment.)

$$D_r(z|x, m) = \sum_{x_1=0}^x \sum_{z_1=0}^z D_{r/2}(z_1|x_1, m) D_{r/2}(z - z_1|x - x_1, m) E_{r/2}(x_1|x_1 + x_2, m) \quad (10)$$

where $E_r(x_1|x_1 + x_2, m)$ is the probability that x_1 cells in the buffer are addressed to output ports in $1(r)$, given that i) $x_1 + x_2$ are addressed to ports in $1(r) \cup 2(r)$ and ii) m sources are active, i.e.,

$$E_r(x_1|x_1 + x_2, m) = \frac{B_r(x_1, x_2, m)}{\sum_{i,j | i+j=x_1+x_2} B_r(i, j, m)}. \quad (11)$$

The knowledge of A_r and D_r permits the formulation of the transition probabilities for any pair of states in the model determined by r , i.e., the model that represents an aggregation of output ports according to sets $1(r)$ and $2(r)$. This, in turn, permits the computation of the steady-state buffer-source distribution B_r , using an iterative method similar to that employed by algorithm 1. This method relies on both A_r and D_r in the sense mentioned above. Accordingly, for a given value of r (i.e., a given iteration of algorithm 2), the calculations of A_r and D_r must precede that of B_r . Once B_r is computed, if $2r = R$ then the computation terminates since, in this case, the r -subsystem accounts for all the output ports. If not, the number of ports considered is doubled (i.e., r is replaced by $2r$) and the computations are repeated for this larger r -subsystem; in particular, the new values for D_r are obtained using the recursion given by (10). For additional details concerning the computation of B_r , please see appendix B. Accordingly, algorithm 2 can be summarized as follows.

$$\text{Step 1: } r = 1. \quad D_1(0|x, m) = \begin{cases} 1 & \text{if } x = 0 \\ 0 & \text{else} \end{cases} \quad \text{and} \quad D_1(1|x, m) = \begin{cases} 1 & \text{if } x > 0 \\ 0 & \text{else} \end{cases}.$$

- Step 2: Compute A_r (see (9)).
 Step 3: Compute B_r (see appendix B).
 Step 4: If $2r = R$, exit; otherwise continue.
 Step 5: $r \leftarrow 2r$.
 Step 6: Compute D_r (see (10)).
 Step 7: Go to Step 2.

This algorithm yields a fairly good approximation of the steady-state occupancy probabilities of a shared-buffer switch with bursty traffic. The two principal sources of approximation error are the following.

1. In solving each r -subsystem model (Step 3), we assume that the storage capacity for cells addressed to the $2r$ ports in $1(r) \cup 2(r)$ coincides with the buffer capacity K . In reality, this capacity is shared among cells destined for all R ports.
2. In the same step, we assume that the number of departures from ports in $1(r)$ is independent of the number of departures from ports in $2(r)$, i.e., for all $t \in T$, the random variables $Z_{1(r),t}$ and $Z_{2(r),t}$ are statistically independent. This is not generally true.

The number of main iterations of this algorithm is clearly $\log_2 R$. However, it's important to note that this number can be reduced if an r -subsystem can be solved directly for a value of r that is greater than 1. This can be done by applying algorithm 1 to a special $N \times r$ shared-buffer system, where sources in the On state transmit cells with probability r/R . Accordingly, Step 1 of (modified) algorithm 2 then begins at a value $r > 1$, where data for this value is supplied by algorithm 1. This combined use of both algorithms (the "integrated" approach referred to in the title and introduction) obviously reduces the execution time. Moreover, it also improves the precision of the results, since each iteration introduces an error of approximation. Further discussion of the nature of such errors is deferred to the end of section that follows.

4 RESULTS

Recalling some of the motivation that was mentioned at the outset (see section 1), because of the severe requirements imposed on ATM cell loss probabilities, highly accurate results (with high levels confidence) are difficult to obtain by simulation. Although there has been some progress in the development of fast simulation techniques for rare events, e.g., various forms of "importance sampling", these typically rely on very special knowledge of the system in question. If approximate analytic methods are used instead then, even though they often provide reasonably accurate results in the higher probability region of buffer occupancy, they tend to be much less accurate for large occupancy values. In other words, the asymptotic behavior of the distribution (its tail) is not well approximated. However, for purposes such as determining buffer dimensions that insure satisfactory loss performance with respect to stringent cell loss probability requirements, accurate knowledge of this tail is crucial.

To pursue this matter in terms of the development of the previous section, we first examine the use of algorithm 1 as it applies to two of the logical output queues of an $N \times N$ shared buffer. Further, we suppose that the capacity of this buffer is large enough so that, effectively,

it can be regarded as infinite. This situation can thus be represented by an (exact) model of an $N \times 2$ shared-buffer switch, where N sources transmit cells with probability $2/N$. To satisfy the “effectively infinite” assumption, we take the buffer capacity K to be large enough to insure cell loss probabilities that are less than 10^{-15} . Due to the small number of output ports, this model can be efficiently solved using algorithm 1. The purpose of the analysis is to study the correlation among the occupancy distributions of two queues in an $N \times N$ system. For the random (Bernoulli) traffic case, the joint occupancy distribution of the two queues was obtained in both [3] (using z -transforms) and [5]. The analysis here is similar to the latter, but is extended to the case of bursty sources.

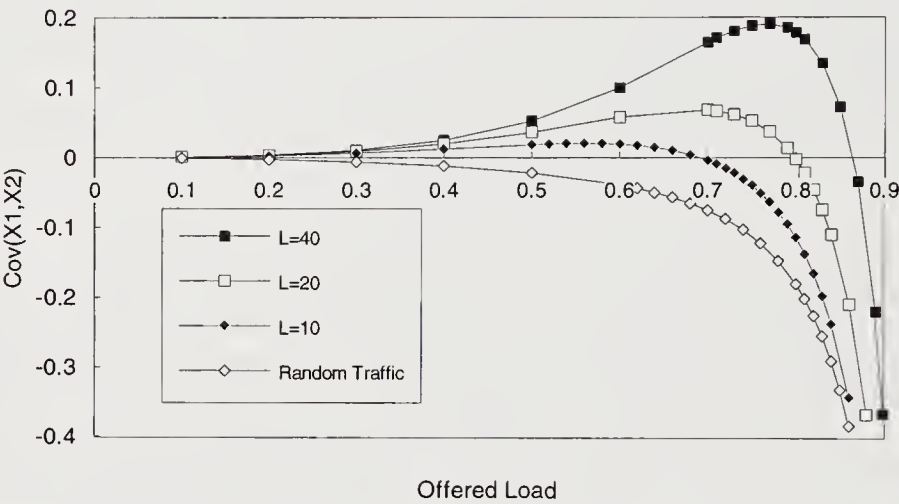


Figure 1 Covariance between the length of two queues.

Figure 1 displays the covariance (between two queues) as a function of offered load, for random and bursty traffic and N equal to 16. Among other things, it is interesting to note the sign of the covariance. For the random traffic case, the covariance is always negative, while in the case of bursty traffic it is positive for low loads and becomes negative as the load increases.

The knowledge of the joint distribution of the occupancy probability of the two queues, together with the assumption of a effectively infinite buffer, permits the variance of the buffer's occupancy distribution to be formulated as follows. Let Y_∞ denote the random variable whose probability distribution is the limiting distribution of Y_t as $t \rightarrow \infty$, i.e., Y_∞ is the steady-state number of cells in the shared buffer. Similarly, let $X_{1,\infty}$ and $X_{2,\infty}$ be the random variables representing the steady-state occupancy of queue 1 and queue 2, respectively, in the $N \times 2$ system described above. Then, taking the subscripts ∞ to be implicit (context should suffice to convey the steady-state interpretation), the mean and variance of $Y = Y_\infty$ can be formulated as follows in terms of $X_1 = X_{1,\infty}$ and $X_2 = X_{2,\infty}$, where we let queue 1 be

representative of single-queue behavior. Note that since X_1 and X_2 are identically distributed, X_2 could likewise serve this purpose.

$$E[Y] = N \cdot E[X_1] \quad (12)$$

$$\text{Var}(Y) = N \cdot \text{Var}(X_1) + N(N-1) \cdot \text{Cov}(X_1, X_2) \quad (13)$$

Formula (13) can also be used to determine the error (with regard to variance) introduced by assuming that the queues are statistically independent. (The latter assumption is convenient since it permits the distribution of Y to be obtained via the N -fold convolution of the distribution of a single queue.) Assuming such independence, $\text{Var}(Y) = N \cdot \text{Var}(X_1)$ and, accordingly, the error due to this assumption is given by $\delta = N(N-1) \cdot \text{Cov}(X_1, X_2)$. Further, since the sign of δ is clearly the sign of the covariance, an approximation based on convolution underestimates the variance if $\text{Cov}(X_1, X_2) > 0$. Since there is no error with regard to the value of $E[Y]$, we can reasonably expect that, by using convolution, the results are conservative only if the covariance is negative. Also, it appears that this approximation gets worse for growing values of $|\text{Cov}(X_1, X_2)|$ and becomes useless when this value approaches that of $N \cdot \text{Var}(X_1)$. This is borne out by Figure 2, which compare occupancy distributions obtained by convolution with corresponding (and more accurate) results determined by simulation. Specifically, this plot demonstrate that convolution provides an overly optimistic estimate of the distribution of Y in the case of heavy, bursty traffic and, hence, cannot be relied on for practical applications.

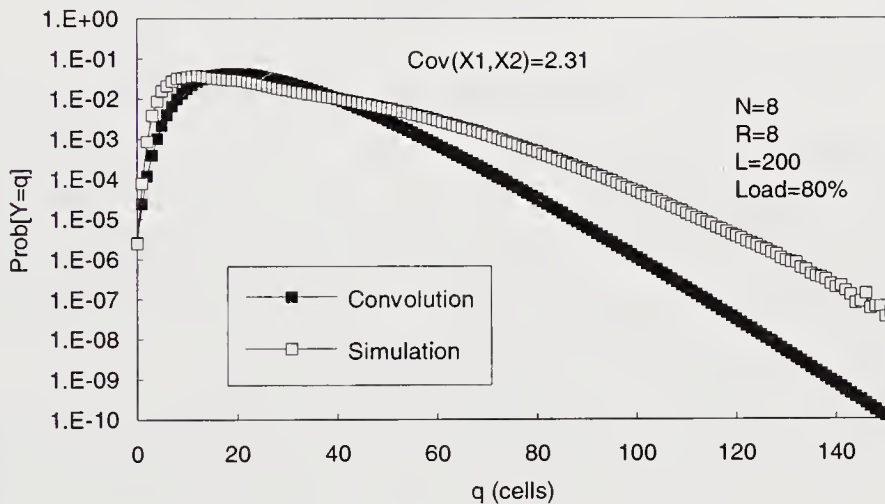


Figure 2: Comparison between convolution and simulation.

It is also worth noting that the knowledge of mean and variance of the shared-buffer occupancy distribution permits another approximation of loss performance. As mentioned in section 1, for the case of a shared buffer submitted to random traffic, [3] has proposed

approximating the occupancy probability of an (infinite) shared buffer with the density of an appropriate Gamma distribution. This particular distribution was considered because of the exponential asymptotic behavior of its density function, which is typical of many queueing systems. More precisely, by computing the mean and variance of the occupancy distribution of an infinite shared buffer, this Gamma distribution is chosen such that its first two moments match the computed values. The loss probability of a K -capacity buffer is then estimated as the probability that a random variable with this Gamma density has a value greater than K .

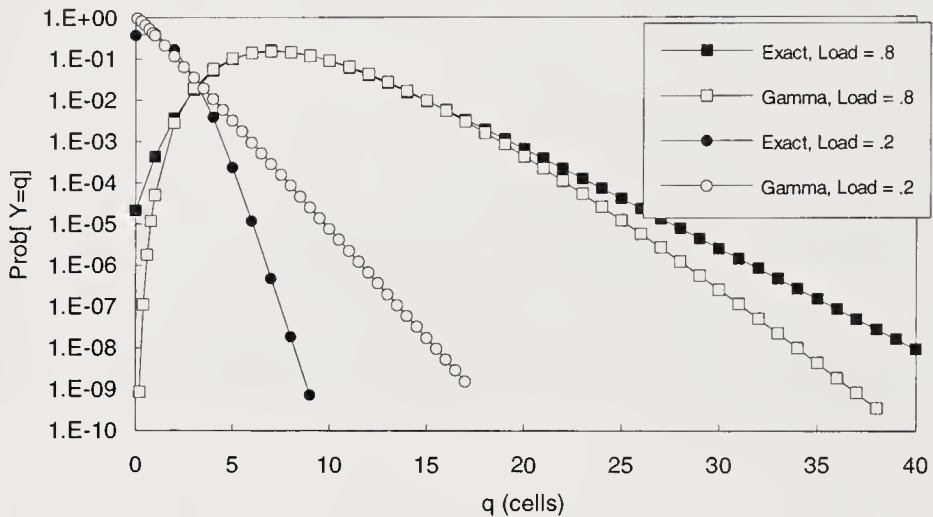


Figure 3: Comparison between exact results and Gamma function approximation.

Figure 3 displays the occupancy distributions of a 4×4 switching element, with $K = 50$ and offered loads ρ equal to 0.2 and 0.8, respectively. The Gamma distribution's density function is then compared with the distribution obtained from algorithm 1, indicating that the Gamma density provides a fairly good approximation for the higher probability states. On the other hand, one can see that it fails to capture asymptotic behavior in the low probability region. Moreover, we see that the estimation errors in this region are load-dependent, with values being overestimated in case $\rho = 0.2$ and underestimated if $\rho = 0.8$.

We now turn to the analysis that utilizes algorithm 2. As noted earlier, this algorithm estimates the loss probability of a shared-buffer system with bursty traffic, even in cases where the buffer is large. To validate this approach, we compare the results obtained by algorithm 2 with those obtained by simulation (for large buffers and high loads) and by algorithm 1 (for smaller buffers).

Figures 4 and 5 plot the loss probability as a function of buffer capacity K for both an 8×8 (Figure 4) and a 16×16 (Figure 5) switch. In each case, we consider random traffic and three different values of offered load ($\rho = 0.2, 0.7, 0.8$).

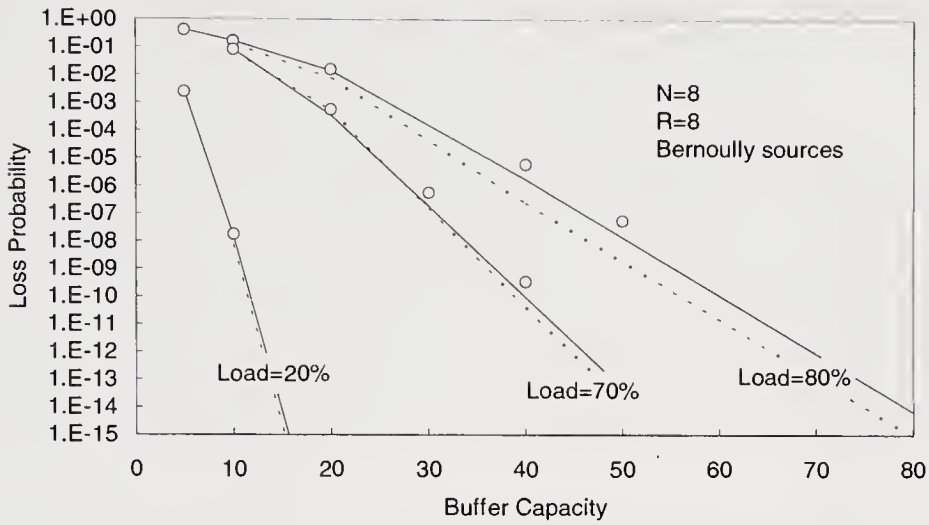


Figure 4: Comparison between exact and approximate results; random traffic.

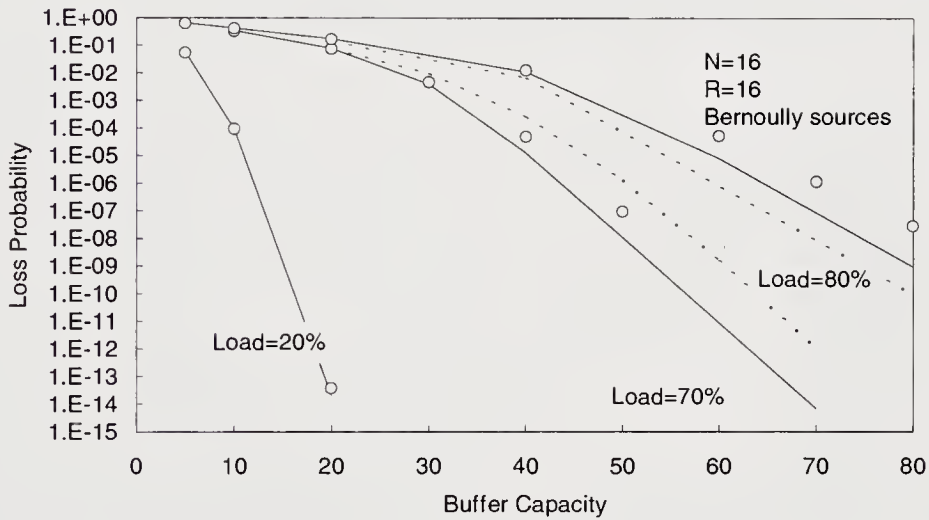


Figure 5: Comparison between exact and approximate results; random traffic.

Comparisons of the model result (straight line) with simulation data and with exact analytic results (both plotted as circles) reveal that the error increases as the capacity K gets larger. However, in all the cases considered, the error's value is less than an order of magnitude. Included in the same figures are some plots of results obtained from the algorithm described

in [5] (dashed lines). In all cases, it can be seen that algorithm 2 provides a more accurate estimate of cell loss probability.

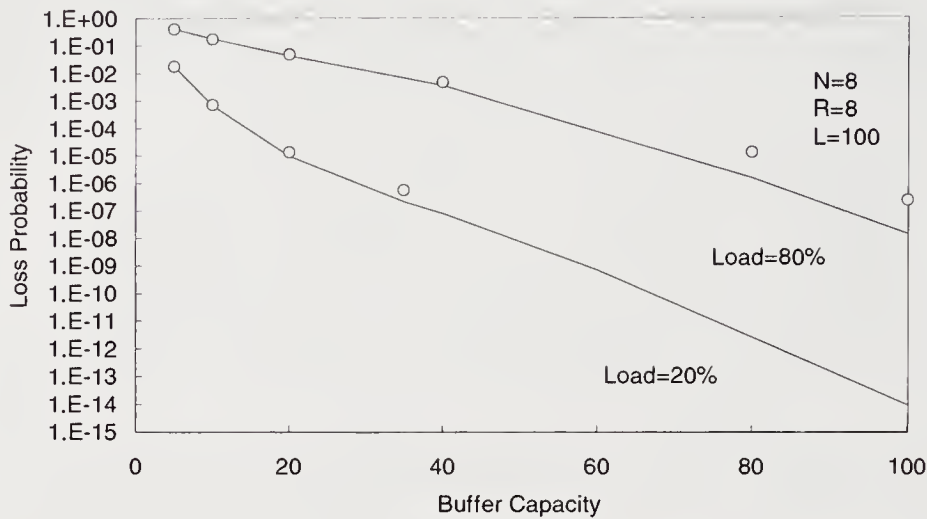


Figure 6: Comparison between exact and approximate results; $L = 100$.

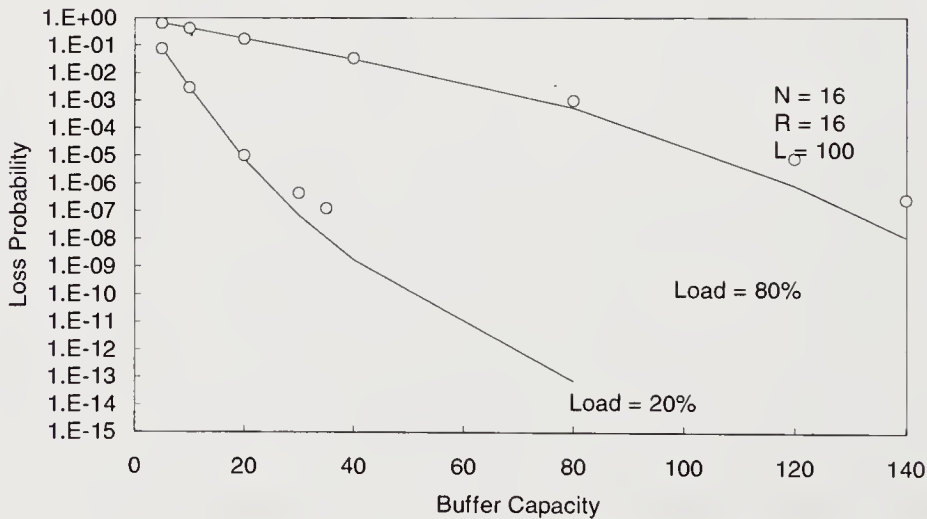


Figure 7: Comparison between exact and approximate results; $L = 100$.

Figures 6 and 7 are similar to Figures 4 and 5, respectively, except that we now consider traffic sources that are bursty. Specifically, the mean burst length of a source is $L=100$ and, again, two instances of offered load are considered, namely $\rho = 0.2$ and $\rho = 0.8$.

To illustrate another application that integrates the use of an exact method (algorithm 1) with an approximate formulation (described below), we estimate the advantages, in terms of required storage capacity, of a shared-buffer architecture as compared with a (simpler) switch having dedicated output queues (no sharing). For given values of N , R , L , and ρ_{in} along with an admissible (maximum allowed) cell loss probability p_a , let K be the capacity of the shared buffer and let K' be the capacity of each of the output buffers in the dedicated case. Hence, RK' is the total capacity of the latter. Further, let s denote the fraction of this capacity that is required in the case of a shared-memory switch (presuming the same value of p_a for each), i.e., $s = K/RK'$. Thus the value of s (the “relative saving”) can vary from a minimum of $1/R$ (corresponding to the greatest theoretical reduction in required memory) to a maximum of 1 (no reduction).

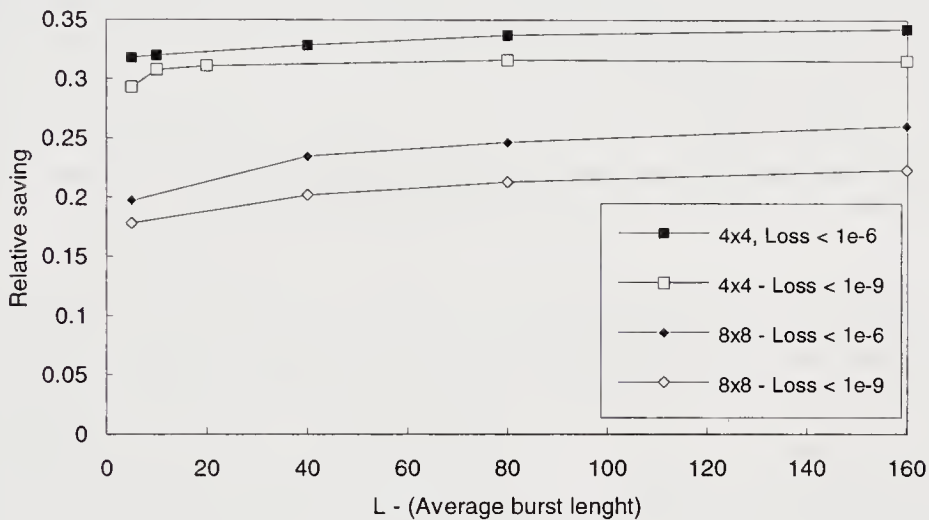


Figure 8: Memory saving afforded by buffer sharing.

Figure 8 illustrates how this relative saving varies as a function of the mean burst length L for different choices of $N = R$ (4, 8). The activity ρ_{in} of a source is equal to 0.8 and two loss probability targets are considered. As indicated by the figure, we have the following observations with respect to the combinations of parameter values considered.

- The advantage of buffer sharing increases (as reflected by smaller values of s) as the number of ports gets larger.
- Likewise, buffer sharing is more advantageous as the loss probability target becomes lower (more severe).

For very low loss probability targets p_a , observation b) suggests an empirical means of obtaining a quick and conservative estimate of the required capacity K of a shared-buffer

switch. Let $K(p_a)$, $K'(p_a)$, and $s(p_a)$ be the values of K , K' , and s that result from a particular choice of p_a . Then, from the definition of s , it follows that

$$K(p_a) = s(p_a)RK'(p_a). \quad (14)$$

If the value of p_a is relatively high (in an ATM context, $p_a \geq 10^{-6}$) then $K(p_a)$ can be obtained by simulation. As for $K'(p_a)$, algorithm 1 can serve to determine its value (letting $R = 1$), even in cases where p_a is much smaller (as we exploit below). Then, to “scale up” these calculations to a more severe loss requirement, let p_a' denote a lower admissible cell loss probability, i.e., $p_a' < p_a$. Then by observation b) it follows that $s(p_a') < s(p_a)$ and hence, applying (14), we have

$$K(p_a') = s(p_a')RK'(p_a') < s(p_a)RK'(p_a') = K(p_a)K'(p_a')/K'(p_a). \quad (15)$$

Given the value of $K'(p_a')$, which again can be accurately determined using algorithm 1, (15) thus provides a conservative estimate of shared-buffer capacity in cases where the target loss probability is much lower (e.g., $p_a' \leq 10^{-9}$), namely

$$K_{\text{est}}(p_a') = K(p_a)K'(p_a')/K'(p_a)$$

For both a 4×4 and an 8×8 switch ($R = 4, 8$) and as a function of traffic burstiness, Figure 9 illustrates the extent to which $K_{\text{est}}(p_a')$ overestimates the actual values.

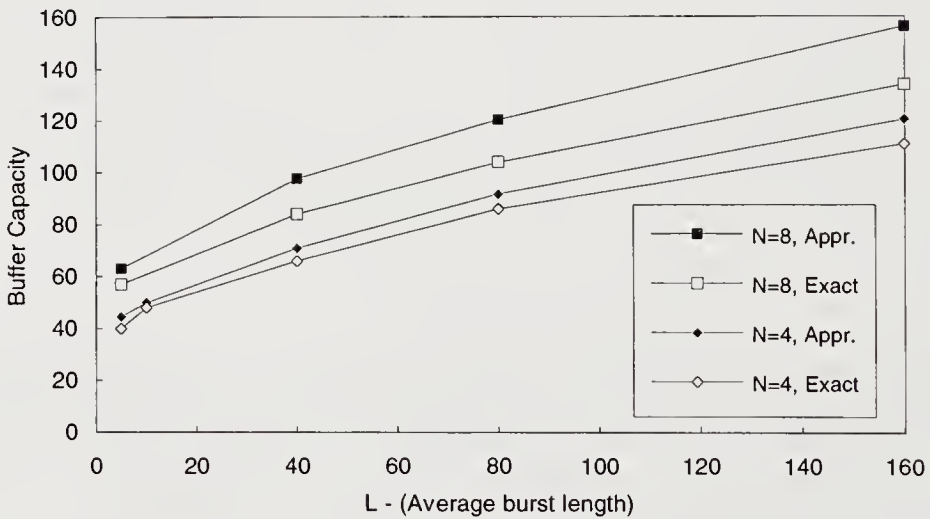


Figure 9: Validation of the empirical dimensioning rule.

Here, the original and modified target loss probabilities considered are $p_a = 10^{-6}$ and $p_a' = 10^{-9}$, respectively, under an assumed offered load of $\rho = 0.8$. As can be observed, the relative error of the estimate (i.e., the quantity $(K_{\text{est}} - K)/K$) is somewhat smaller for $R = 4$ as compared with $R = 8$. For both switches, this error increases slowly as the mean burst length L becomes larger. In particular, for an 8×8 switch with $L = 160$, the relative error is approximately 0.2 (20%).

5 SUMMARY

By employing an exact solution method (algorithm 1) along with a new approximate method (algorithm 2), we have shown that a synergistic use of both can be beneficial in the context of shared-buffer switch evaluation. Aside from the usual advantages associated with comparing exact vs. approximate results, we find that true integration, where both are employed for a single purpose, can likewise be very useful. Further, we have shown that algorithm 2, particularly if used in concert with algorithm 1, can provide reasonably accurate results even for realistically large switches in the presence of a bursty traffic environment. Finally, by examining the reduction in buffer capacity, relative to a dedicated-buffer architecture, that results from buffer sharing, we have found that memory saving increases as the target loss probability decreases. In turn, this suggested an empirical means of conservatively dimensioning a shared-buffer switch such that, even in the case of extremely low target probabilities, the capacity so determined is reasonably close to what's actually required.

6 REFERENCES

- [1] Devault, M., Cochenne, J. and Servel, M. (1988) The PRELUDE ATD experiment: Assessments and future prospects. *IEEE Journal on Selected Areas in Communications*
- [2] Hluchyj, M. and Karol, M. (1988) Queueing in high performance packet switching. *IEEE Journal on Selected Areas in Communications*
- [3] Eckberg, A.E. and Hou, T.C. (1988) Effects of output buffer sharing on buffer requirements in an ATDM packet switch, in *Proc. INFOCOM '88*, New Orleans, LA.
- [4] Bianchi, G. and Turner, J., (1993) Improved queueing analysis of a shared buffer switching network, in *Proc. INFOCOM '93*, San Francisco, CA.
- [5] Meyer, J. F., Montagna, S. and Paglino, R. (1993) Dimensioning of an ATM switch with Shared Buffer and Threshold Priority. *Computer Networks and ISDN Systems*, **26**, 95-108
- [6] Louvion, J., Boyer, P. and Gravey, A. (1988) A discrete time single-server queue with Bernoulli arrivals and constant service time, in *Proc. 12th Int'l Teletraffic Congress*, Torino, Italy
- [7] Petit, H. and Desmet, M. (1990) Performance evaluation of shared buffer multiserver output queue switches used in ATM, in *Proc. 7th ITC Seminar*, Morristown, NJ
- [8] Yamashita, H., Perros, H.H. and Hong, S.W. (1991) Performance modeling of a shared buffer ATM switch architecture, in *Proc. 13th Int'l Teletraffic Congress*, Copenhagen, Denmark

APPENDIX A

Recalling notation introduced in section 3.1, we let Q be the set of possible (buffer occupancy) states of the R logical queues of a shared buffer of finite capacity K , i.e., if q_i is the number of cells stored in logical queue i ($0 \leq q_i \leq K$, $1 \leq i \leq R$) then

$$Q = \{(q_1, q_2, \dots, q_R) \mid 0 \leq q_1 + q_2 + \dots + q_R \leq K\}.$$

In turn, we identify states that are permutations of one another via an equivalence relation on Q , letting \bar{Q} denote its corresponding partition (set of equivalence classes). As noted in section 3.1, the shared buffer can then be represented by the reduced stochastic process $\bar{X} = \{\bar{X}_t \mid t \in T\}$, where \bar{X}_t is the equivalence class that contains state X_t .

In what follows, we show how the transition structure of X can be described, in part, by a convenient representation of the elements of \bar{Q} . Specifically, if $q = (q_1, q_2, \dots, q_R) \in Q$, let \bar{q} denote its equivalence class, i.e., the state in \bar{Q} that contains q . Then an *occupancy value* for \bar{q} is the value of some coordinate q_i of q . If, further, we let $b = (b_1, b_2, \dots, b_r)$ be a listing, in increasing order, of all the different occupancy values for \bar{q} (where $1 \leq r \leq R$), we can define the *occupancy vector* of \bar{q} to be the r -tuple $e = (e_1, e_2, \dots, e_r)$, where, e_j is the number of different logical queues having occupancy value b_j ($1 \leq e_j \leq R$). Note that, by the definitions of \bar{Q} and \bar{q} , it follows that of both b and e are invariant relative to the choice of a representative state $q \in \bar{q}$. It is also easily shown that if \bar{q} and \bar{q}' are distinct states in \bar{Q} then the corresponding ordered pairs (b, e) and (b', e') are likewise distinct. In other words, this pair of r -tuples provides a unique representation of a state in \bar{Q} . Moreover, the set of all such representations is just the set of all ordered pairs (b, e) , with $b = (b_1, b_2, \dots, b_r)$ and $e = (e_1, e_2, \dots, e_r)$, such that

$$\begin{aligned} 1 &\leq r \leq R, \\ 0 &\leq b_i \leq K, \\ b_1 &< b_2 < \dots < b_r, \\ \sum_{j=1}^r e_j &= R, \text{ and} \\ \sum_{j=1}^r b_j e_j &\leq K \end{aligned}$$

From this point on, a state of the process \bar{X} will be identified with its corresponding pair (b, e) , where the latter is now regarded as an element of \bar{Q} . In these terms, the transition structure of the process can be formulated as follows.

Given that the system is in a state $\bar{X}_t = (b, e)$ at the beginning of time slot t , the state of the system after departures caused by the "send" operation in slot t (this is an intermediate state that precedes the subsequent arrivals; hence, it is not an explicit part of the behavior of \bar{X}) is given by the function $d(b, e)$, where

$$d(b, e) = \begin{cases} ((b_1 - 1, b_2 - 1, \dots, b_r - 1), e) & \text{if } b_1 > 0 \\ ((b_1, b_2 - 1, \dots, b_r - 1), e) & \text{if } b_1 = 0 \text{ and } b_2 > 1 \\ ((0, b_3 - 1, \dots, b_r - 1), (e_1 + e_2, e_3, \dots, e_r)) & \text{if } b_1 = 0 \text{ and } b_2 = 1 \end{cases}$$

Suppose now that (b, e) is the intermediate state so determined by the function d . The next state \bar{X}_{t+1} is then a function of the arriving cells as well as (b, e) . Assuming that each cell is randomly addressed to one of the output queues, the probability that an incoming cell is addressed to one of the e_j logical queues (each with b_j cells) is clearly e_j/R . In a manner similar to how d is defined, a function a (suggesting “arrival”) then determines an intermediate state (unless it’s the last arrival during slot t) that results from an entry of an arriving cell to one of these e_j queues. Specifically, the function a (whose arguments are the intermediate state (b, e) , along with the value j that identifies the queue subset containing the arrival’s destination queue) can be expressed as follows.

$$a((b, e), j) = \begin{cases} ((b_1, \dots, b_j, b_j + 1, b_{j+1}, \dots, b_r), (e_1, \dots, e_{j-1}, e_j - 1, e_{j+1}, \dots, e_r)) & \text{if } b_{j+1} > b_j + 1 \text{ and } e_j > 1 \\ ((b_1, \dots, b_{j-1}, b_j + 1, b_{j+1}, \dots, b_r), (e_1, \dots, e_{j-1}, 1, e_{j+1}, \dots, e_r)) & \text{if } b_{j+1} > b_j + 1 \text{ and } e_j = 1 \\ (b, (e_1, \dots, e_{j-1}, e_j - 1, e_{j+1} + 1, e_{j+2}, \dots, e_r)) & \text{if } b_{j+1} = b_j + 1 \text{ and } e_j > 1 \\ ((b_1, \dots, b_{j-1}, b_{j+1}, \dots, b_r), (e_1, \dots, e_{j-1}, e_{j+1} + 1, e_{j+2}, \dots, e_r)) & \text{if } b_{j+1} = b_j + 1 \text{ and } e_j = 1 \end{cases}$$

As described in section 3.1, these functions then serve to formulate the transition probabilities of the composite process $Z = \{(\bar{X}_t, M_{t+1}) \mid t \in T\}$.

APPENDIX B

The key step of the approximate method proposed in section 3.2 consists of finding the steady-state distribution $B_r(x_1, x_2, m)$, given the distributions $D_r(z \mid x, m)$ and $A_r(w_1, w_2 \mid m)$ of the departure and arrival processes, respectively. Recalling the exact meanings of each, we have

- $B_r(x_1, x_2, m)$ = the steady-state probability of having x_1 cells in the buffer addressed to output ports in the set $1(r) = \{1, 2, \dots, r\}$, x_2 addressed to output ports in the set $2(r) = \{r+1, r+2, \dots, 2r\}$, and m sources in the On state.
- $D_r(z \mid x, m)$ = the steady-state probability of having z cell departures during a slot from ports in $1(r)$ (alternatively $2(r)$), given that x cells are addressed to these ports and m sources are active.
- $A_r(w_1, w_2 \mid m)$ = the steady-state probability of having w_1 cell arrivals during a slot addressed to ports in $1(r)$ and w_2 addressed to ports in $2(r)$, given that m sources are active.

To compute $B_r(x_1, x_2, m)$, we employ an iterative algorithm similar to the one used for the exact model. Let $B_r^t(x_1, x_2, m)$ be the probability distribution, at time t , which, in the limit (as

$t \rightarrow \infty$), yields the steady-distribution $B_r(x_1, x_2, m)$. Further, let $C_r^t(x_1, x_2, m)$ be the probability distribution of the intermediate state, during slot t , that results from cell departures (but is prior to slot t arrivals). Then it can be shown that, for all $t \in T$,

$$C_r^t(x_1, x_2, m) = \sum_{i=x_1+\min(x_1,1)}^{\min(x_1+r,K)} \sum_{j=x_2+\min(x_2,1)}^{\min(x_2+r,K-i)} B_r^t(i, j, m) D_r(i-x_1|i, m) D_r(j-x_2|j, m)$$

$$B_r^{t+1}(x_1, x_2, m) = \sum_{i=\max(x_1-m,0)}^{x_1} \sum_{j=\max(x_2-m+x_1-i,0)}^{x_2} \sum_{l=0}^N C_r^t(i, j, l) A_r(x_1-i, x_2-j|m) S(ml|l)$$

where $S(ml|l)$ is the (time-invariant) probability that m sources are active in a slot, given that l were active in the previous slot. As earlier, we then iteratively compute $B_r^t(x_1, x_2, m)$ for growing t until we reach a time t that yields a sufficiently close approximation of the steady-state distribution $B_r(x_1, x_2, m)$. Again, the specific criterion for termination is a value of t such that the maximum relative difference between the probabilities for slots t and $t+1$ is less than some very small number.

Sergio Montagna was born in Italy in 1955. He received a degree in physic from the University of Pavia (Italy) in 1978. He joined the Central Research Laboratories of Italtel in January 1983. His current research is concerned in the performance evaluation of ATM systems and networks.

Roberto Paglino was born in Italy in 1962. He graduated from the Università Statale di Milano (Milan, Italy), where he obtained a degree in computer science. Since 1987 he has been with Italtel as a researcher in the sytem engineering department of the central R&D laboratories. His interest are in the performance and architecture of ATM systems.

John F. Meyer received the B.S. degree from the University of Michigan, Ann Arbor, the M.S. degree from Stanford University, Stanford, CA, and the Ph.D. degree in communication sciences from the University of Michigan in 1957, 1958, and 1967, respectively. He is currently a Professor in the Department of Electrical Engineering and Computer Science at the University of Michigan. He has been active in computer and system research for 35 years and has published widely in the areas of fault-tolerant computing and model-based evaluation of system performance, dependability, and performability.

Strictly Nonblocking Operation of 3-stage Clos Switching Networks

Fotios K. Liotopoulos and Suresh Chalasani

University of Wisconsin-Madison

Dept. of Electrical and Computer Engineering, Univ. of

Wisconsin-Madison, 1415 Engineering Dr., Madison, WI 53706, USA.

Telephone: (608)-265-2639. Fax: (608)-262-1267.

Email: {fotios,suresh}@ece.wisc.edu

Abstract

This paper studies the nonblocking switching operation of 3-stage Clos networks in the multirate environment. In particular, we concentrate on the *strictly nonblocking* mode of operation of these switching networks. Our analysis determines bounds for the minimum number of middle-stage switches required for strictly nonblocking operation. Several cases of the multirate environment are considered, including discrete and continuous bandwidth multirate traffic. We survey the results already reported in the literature and we extend them for general asymmetrical Clos networks. We also generalize them for the case in which the internal links have higher bandwidth capabilities than the input/output ports. In addition to these extensions, we derive a more general result, which not only provides a tight bound for various multirate cases, but also improves an already existing bound for the case of continuous bandwidth multirate traffic.

Keywords

Asymmetrical Clos networks, nonblocking operation, multirate networks, computer communication networks, ATM switches.

1 INTRODUCTION

The theory of nonblocking switching was originally motivated by the problem of designing switching systems capable of connecting any pair of idle ports under arbitrary traffic conditions.

One of the first switching fabrics that was recognized to achieve nonblocking operation was the crossbar switch with n input/output ports and n^2 crosspoints, but at a prohibitive cost for large systems. In 1953, Charles Clos (Clos,1953) wrote a famous paper introducing design methodologies for switching networks capable of achieving nonblocking operation with significantly less crosspoints. This milestone work was the foundation of the theory

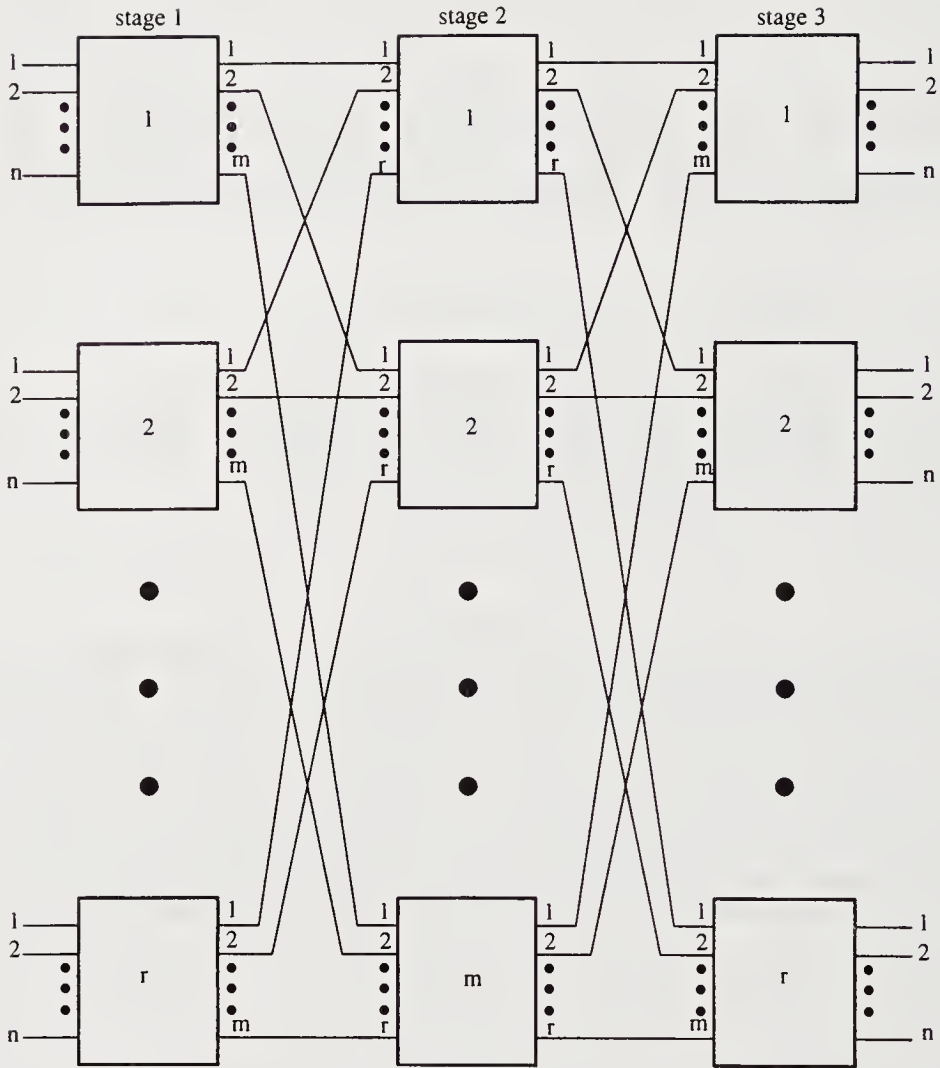


Figure 1 A symmetric 3-stage Clos network.

that has since been developed by Beneš, Pippenger and many others, *e.g.* (Beneš,1965), (Pippenger,1982), (Cantor,1971), (Feldman *et al.*,1986), and (Masson *et al.*,1979).

1.1 General Description of Clos Networks

A three-stage Clos network consists of three successive stages of switching elements which are interconnected by links. In a symmetric three-stage network, all switching elements in a stage are uniform (see Figure 1). In the symmetric Clos network of Figure 1, there are r switches of size $n \times m$ in the first stage, m switches of size $r \times r$ in the second, and r switches of size $m \times n$ in the third. This network thus interconnects $n \cdot r$ input ports with $n \cdot r$ output ports.

In this paper, we also consider the more general class of asymmetrical 3-stage Clos networks (Varma *et al.*,1993) and study their nonblocking operation in the *Multirate* environment.

Figure 2 illustrates the general class of asymmetrical 3-stage Clos networks considered in this paper. The differences from the symmetric case are in the number of switches per stage, the number of input/output ports per switch and in the capacities of the internal links. In the asymmetrical case, all these quantities may have different values.

1.2 Multirate Networks

In the early years of the exploration of switching networks, the control algorithms to achieve nonblocking operation were simple. The method of Circuit Switching was the dominant control operation, according to which, in order to establish a connection from point A to point B, one had to progressively reserve the links and establish the entire path from A to B. All the resources in that path were allocated to the connection during its lifetime, independent of the connection's bandwidth requirements.

In this Classic Circuit Switching (CCS) environment, the fundamental assumption is that each link is allocated to exactly one connection, That is, no link can accommodate more than one connection simultaneously.

Since those times, circuit switching methods have evolved significantly. Physical links are now time-multiplexed and the concepts of virtual channels and packet switching have matured and have been applied extensively.

During the past 15 years, the trend in communication systems has been towards serving applications with a wide variety of characteristics. Such systems are designed to support connections with arbitrary data rates, ranging from a few bits per second to several hundreds of megabits per second, *e.g.* (Coudreuse *et al.*,1987), (Huang *et al.*,1984), and (Turner,1988). These systems usually carry information in multiplexed format. However, in contrast to earlier systems, each connection can utilize an arbitrary fraction of the bandwidth of the link carrying it. Typically, information is carried through a link by statistically multiplexing a set of data-packet sequences belonging to a number of services. The general class of switching networks that can support services with arbitrary bandwidth requirements are called *multirate networks*. Multirate networks can be considered as the infrastructure to support the Asynchronous Transfer Mode (ATM) protocol for broadband integrated services (B-ISDN).

Both constant bit-rate (CBR) services as well as variable bit-rate (VBR) services can be accommodated in the multirate switching environment. A VBR service can be considered as a sequence of CBR services, each of which has a fixed bandwidth requirement during a specific time interval. This bandwidth requirement is the maximum bandwidth requirement of the corresponding VBR service throughout the same time interval. Bursty traffic can be modeled in a similar way as traffic consisting of VBR services. Therefore, a service with bursty characteristics may be connected, disconnected, or rearranged at different time intervals, depending on its bandwidth demands.

One way to operate multirate networks is to select, for each incoming connection, a path through the switching system to be used by all packets belonging to that connection. The path selection should be such that the available bandwidth on all intermediate links selected is enough to carry the connection through. In order to guarantee high-quality

of service, our goal should be to design switching networks with as small a blocking probability as possible.

Multirate switching provides the framework for popular trends such as *multimedia* and highly diverse service demands, as well as for state-of-the-art standards such as the *ATM* (Asynchronous Transfer Mode). Therefore, it is very important to study and explore the properties of nonblocking operation of multirate switching systems. Such a study is also necessary because the multirate model further covers *packet-switching* techniques, which cannot be analyzed by the CCS model. The analysis of traditional circuit switching methods provides us with the tools to use in today's demanding B-ISDN * environment.

The extension to multirate switching is not a trivial problem. The primary difficulty arises from the fact that the bandwidth requirements of the various services are not fixed any more, but instead can be very diverse. More sophisticated algorithms are needed, in order to allocate the required bandwidth for each service request appropriately, without compromising hardware resources or the quality of service. These complex algorithms, along with the variability of the service demands, make the analysis of multirate networks a very challenging problem.

We consider two cases of the multirate environment in our study: (a) the *discrete bandwidth* case, and (b) the *continuous bandwidth* case.

Discrete Bandwidth case: The weight of all connections belongs to a finite set $\{b_1, \dots, b_K\}$, where $b_k = kb_1$, $k = 1, \dots, K$. Denote $b = b_1$ and $B = \max_k \{b_k\}$. In this case, in order to simplify the notation, we will assume that $1/b$ is an integer, although the proofs still hold (with a little modification) if we do not impose this restriction.

Continuous Bandwidth case: Assume that each input and output port has a maximum capacity of β , and each internal link has a maximum capacity of 1, (b, B, β are normalized with respect to the internal link capacity). Then, the weights of all connections belong to the closed interval $[b, B]$, where $0 \leq b \leq B \leq \beta \leq 1$.

The analysis for the continuous bandwidth case can also be used for the discrete bandwidth case, since the discrete bandwidth case is a special case of the continuous bandwidth case. However, the discrete bandwidth case is more restrictive and, therefore, tighter bounds can be derived for it by using specialized analysis. Furthermore, this special case arises in several applications, which do not require the generality of the continuous bandwidth environment. For these reasons, we study both cases separately.

1.3 Nonblocking Operation

One way to operate switching networks is to select for each incoming connection a path through the switching system to be used by all packets belonging to that connection. The path selection should be such that the available bandwidth on all intermediate links selected is enough to carry the connection through. In order to guarantee high quality of service, our goal should be to design switching networks with as small a blocking probability as possible. Therefore, it is very important to study and explore the properties of nonblocking operation of multirate switching systems.

There are more ways than one for a switch to achieve nonblocking operation, based on

*Broadband Integrated Services Digital Networks

Table 1 Overview of the related work.

Type	Conditions	if/iff	Formula	Authors
CCS	$b = B = \beta = 1$	iff	$M \geq 2L - 1$	Clos
DBW	$\frac{\omega}{b}, \frac{1}{b}$ int.	iff	$M \geq 2 \cdot \left\lfloor \frac{L-B}{1-B+b} \right\rfloor + 1$	Chung & Ross
CBW	$b \leq 0.5$	if	$M \geq 2 \cdot \max_{\omega \in [b, B]} \left\lfloor \frac{\beta L - \omega}{\max\{1-\omega, b\}} \right\rfloor + 1$	Melen & Turner
CBW	$b > 0, B \in (1 - b, 1]$	iff	$M \geq 2 \cdot \left\lfloor \frac{1}{b} \right\rfloor \cdot (L - 1) + 1$	Chung & Ross
CBW	$b = 0, B \in (0, 1)$	iff	$M \geq 2 \cdot \left\lceil \frac{L-1}{1-B} \right\rceil + 1$	Chung & Ross
CBW	$\frac{B}{b}, \frac{1}{b}$ int.	if	$2 \left\lfloor \frac{L-B}{1-B+b} \right\rfloor + 1 \leq M^* \leq 2 \left\lceil \frac{L-B}{1-B} \right\rceil + 1$	Chung & Ross
CCS: Classic Circuit Switching, DBW/CBW: Discrete/Continuous BandWidth				

the way the incoming connections are distributed over the network’s resources. Each connection can be established by allocating network resources (links and buffers) in a number of ways, ranging from naive, arbitrary-allocation to highly sophisticated and complex algorithms. Methods of higher complexity usually achieve better nonblocking performance at lower cost.

1.4 Earlier Work

Table 1.4 summarizes the results that have been reported in the past for these networks and for both discrete and continuous bandwidth multirate traffic. The formulas included in this table provide bounds for the minimum number of middle-stage switches required for strictly nonblocking (SNB) operation.

In 1953, Charles Clos (Clos,1953) introduced for the first time the class of 3-stage Clos networks and determined that the number of middle-stage switches M , that are necessary and sufficient for strictly nonblocking operation, must be at least $2L - 1$, where L is the number of input ports per first-stage switch. His result applies in the most general case of the classic circuit switching (CCS) environment, i.e., for $b = B = \beta = 1$. The proof of this result relies on the fact that, in the worst case, the source input switch can have up to $L - 1$ input ports occupied, and each of the corresponding connections can be routed through a different middle-stage switch, thus requiring $L - 1$ such switches. For the same reason, the destination output switch may require $L - 1$ additional (possibly different) middle-stage switches. Therefore, in order to be able to route a new connection we need $(L - 1) + (L - 1) + 1 = 2L - 1$.

In 1983, Jajszczyk (Jajszczyk,1983) derived conditions for nonblocking operation in the strict sense of two and three stage, multiple channel networks.

In 1989, Melen and Turner (Melen *et al.*,1989) introduced an elegant model for interconnection networks that also considers multirate traffic. Their analysis determined a lower

bound for the continuous bandwidth multirate environment with $b \leq 0.5$, (for $b > 0.5$ we have the CCS environment case). In the multirate environment, the main difference is that, now, more than one connections may share a link's bandwidth. Therefore, the study and analysis of multirate networks involves considering the available bandwidth instead of available links.

In 1991, Chung and Ross (Chung *et al.*, 1991) improved on the bounds derived by Melen and Turner, by deriving sufficient and necessary conditions for various cases of continuous bandwidth multirate environments, as well as for one case of discrete bandwidth multirate environment, (see Table 1.4).

In 1994, Collier and Curran (Collier *et al.*, 1994) generalized Melen and Turner's conditions for three stage, multirate networks, including asymmetrical switch configurations.

Chung and Ross, although they produced tighter bounds and conditions, they only considered the special case, where the i/o ports have the same capacity as the internal links (i.e., equal to 1). Melen and Turner, instead, had considered the more general case of the internal links having a different (higher) capacity than the i/o ports. Our work extends the results of Chung and Ross, based on this assumption by Melen and Turner. In addition to this, we derive more results for cases not covered so far, such as the case for $B < 1 - b$.

The general methodology for deriving these results is by considering one first-stage switch (source) and one third-stage switch (destination), and assume the worst case scenario, as outlined above.

The rest of this paper is organized as follows. Section 2 introduces the necessary notation and definitions. Section 3 presents a number of theorems and corollaries which generalize earlier work for various cases of the multirate environment.

2 PRELIMINARIES

In this section, we present the notation that will be used and referred to throughout this paper. We also define the strictly nonblocking mode of switching operation.

Figure 2 illustrates the general class of asymmetrical three-stage Clos networks considered in this paper. The number of switching elements in the three stages are F , M , and G , respectively. We will use the notation $S1_i$, $S2_j$, $S3_k$ to refer to first-stage switch i , middle-stage switch j and third-stage switch k , respectively, for $i = 1, \dots, F$, $j = 1, \dots, M$ and $k = 1, \dots, G$. The interconnection pattern among the three stages is described below.

Stage 1: Switching element $S1_i$ has P_i input ports. It is connected to switching element $S2_j$ with a link of capacity $R_{i,j}$. The link between $S1_i$ and $S2_j$ is denoted as $L1_{i,j}$.

Stage 3: Switching element $S3_k$ has Q_k output ports. It is connected to switching element $S2_j$ with a link of capacity $T_{j,k}$. The link between $S2_j$ and $S3_k$ is denoted as $L2_{j,k}$.

The capacities $R_{i,j}$ and $T_{j,k}$ are normalized quantities such that:

$$\min \left\{ \min_{i,j} R_{i,j}, \min_{j,k} T_{j,k} \right\} = 1.$$

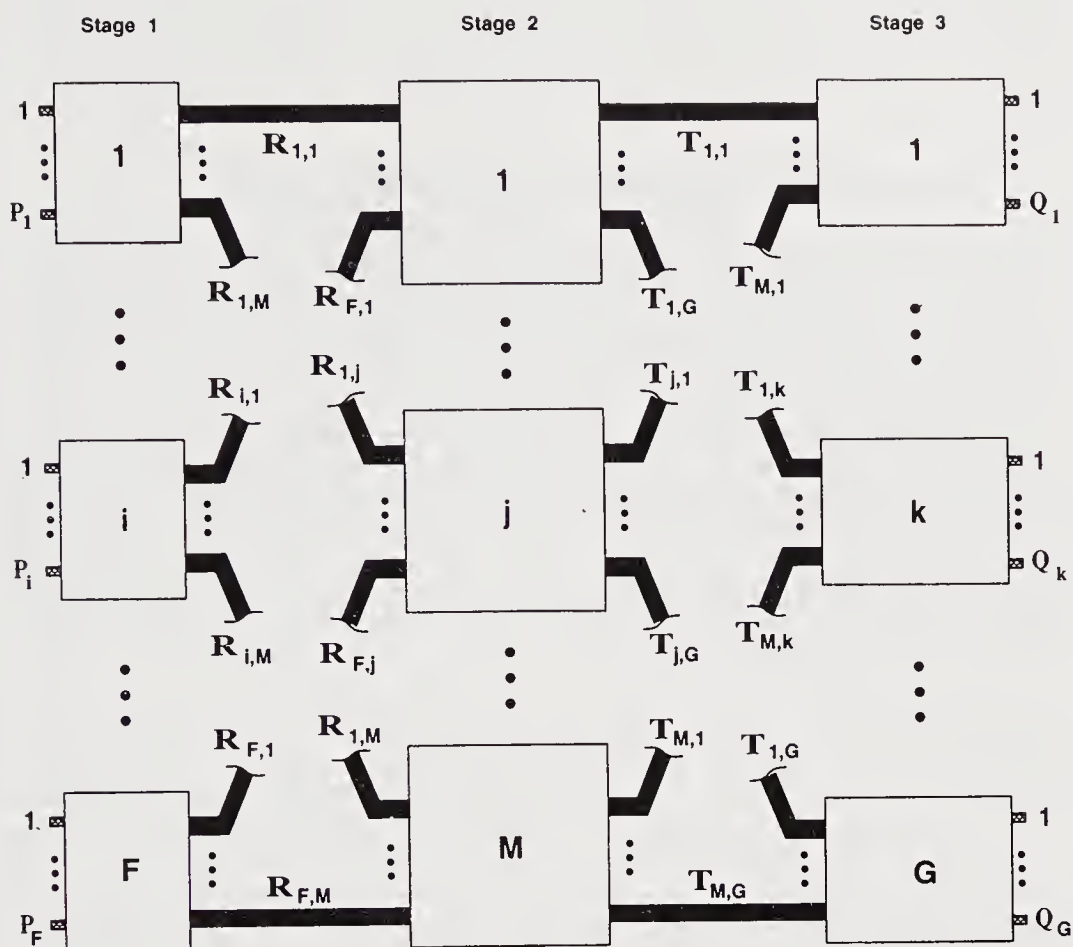


Figure 2 A general asymmetrical Clos network.

Each input or output link has a (normalized) capacity of $\beta \leq 1$. Therefore, the total capacity of all the input ports to the network is given by $\mathcal{I} \stackrel{\text{def}}{=} \sum_{i=1}^F P_i \beta$. Similarly, the total capacity of all output ports of the network (i.e. outputs of the third stage) is $\mathcal{O} \stackrel{\text{def}}{=} \sum_{k=1}^{k=G} Q_k \beta$. The total link capacity, \mathcal{C} , from the first stage to the middle stage is given by

$$\mathcal{C} \stackrel{\text{def}}{=} \sum_{i=1}^F \sum_{j=1}^M R_{i,j}.$$

Similarly, the total link capacity, \mathcal{D} , between the middle and the third stages is given by

$$\mathcal{D} \stackrel{\text{def}}{=} \sum_{j=1}^M \sum_{k=1}^G T_{j,k}.$$

Therefore, the total capacity of any set of simultaneous connections in the network cannot exceed

$$\min\{\mathcal{I}, \mathcal{C}, \mathcal{D}, \mathcal{O}\}.$$

2.1 Definition of the Strictly Nonblocking Mode

Before we define the strictly nonblocking mode of operation of 3-stage Clos networks, we will introduce the required notation for this definition. For this purpose, we will consider the more general multirate environment, since the classic circuit switching (CCS) definitions can then be obtained as special cases. Most of the notation presented in this section is taken from (Melen *et al.*, 1989).

In the multirate environment, we assume that each connection request has a weight $\omega \in [b, B]$ for some $b \leq B$. A multirate network is said to operate under the *continuous bandwidth* assumption, if ω can take any real value in the interval $[b, B]$. If ω can only take a few discrete values in $[b, B]$ then the network is said to operate in the *discrete bandwidth* case. A *route* of weight ω is a sequence of links forming a path from an input port to an output port such that a bandwidth of ω is allocated on each link of this path. A route of weight ω from x to y *realizes* a request $[x \rightarrow y, \omega]$. A *state* is a set of routes that satisfy all the following conditions.

1. For every input or output port x , the sum of the weights of the routes that include x is at most β .
2. For every link $L_{1,i,j}$, the sum of the weights of all routes that use $L_{1,i,j}$ is at most $R_{1,i,j}$.
3. For every link $L_{2,j,k}$, the sum of the weights of all routes that use $L_{2,j,k}$ is at most $T_{j,k}$.

We say that a state realizes a set of requests, if there is a one-to-one and onto mapping from the set of requests to the set of routes in the state. Notice that the utilization of a link l in a given state is the sum of the weights of all routes that include l . A link or switch y is said to be ω -*accessible* in a given state from an input port x , if there is a path from x to y such that the weight on each link l in the path is at most $C_l - \omega$, where C_l is the capacity of the link l . We say that a connection request $[x \rightarrow y, \omega]$ is *compatible* with a state s , if the weight on x and y in s is at most $\beta - \omega$.

Based on the above definitions, we can now define the Strictly Nonblocking (SNB) mode of operation as follows.

Strictly Nonblocking Operation (SNB): A network is strictly nonblocking, if for every state s and for every connection request r compatible with s , there exists a route realizing r in s . No specific control algorithm is assumed and no rearrangements of existing connections may be performed.

Although there are three more nonblocking modes defined for 3-stage Clos switching networks,[†] in this paper, we concentrate on the analysis of the strictly nonblocking mode

[†]Wide-Sense Nonblocking (WSN), Semi-Rearrangeably Nonblocking (SRN), and Rearrangeably Nonblocking (RNB) mode. Only SRN and RNB involve rearrangements of existing connections.

of operation. Since the SNB mode is the most general one, all the results derived for the SNB mode are also sufficient conditions for the other three nonblocking modes.

3 THEORETICAL ANALYSIS FOR SNB OPERATION

All the theoretical results presented in this section refer to the Strictly Nonblocking (SNB) mode of operation of 3-stage Clos networks. Recall that in this mode connections are established and disconnections are performed without any particular algorithm. We derive results for both discrete and continuous bandwidth cases.

Observe that *if a given interconnection network is SNB for the continuous bandwidth case, it is also SNB for the discrete bandwidth case*. Studying the special case (i.e., the discrete bandwidth case) separately allows us to obtain better bounds, which are still useful in applications where the discrete bandwidth assumption is appropriate and adequate. Also, note that the CCS case corresponds to $b = B = 1$ in the discrete bandwidth case.

Throughout this section, we will adopt the original notation used by Chung & Ross (Chung *et al.*, 1991).

For the following analysis, assume a 3-stage Clos network with N input and N output ports. Each first-stage switch has L inputs M outputs and each third-stage switch has M inputs and L outputs. Therefore, we have a $\frac{N}{L} \times M \times \frac{N}{L}$ Clos switch (see Figure 3).

In the multirate environment, recall that each connection request has a weight $\omega \in [b, B]$ for some $b \leq B$. Also, each input and output port has a maximum capacity of β , and each internal link has a maximum capacity of 1 (b, B, β are normalized with respect to the internal link capacity).

Let M^* be the minimum number of middle-stage switches for the 3-stage Clos network to be strictly nonblocking (with N and L fixed). It is well-known from (Beneš, 1965) that $2L - 1$ middle-stage switches are required for strictly nonblocking operation in the CCS case. The following theorems determine the number of middle-stage switches required for strictly nonblocking operation in the multirate environment, both in the discrete and the continuous bandwidth cases.

3.1 Already existing results

The following theorems present the results summarized in Table 1.4, and refer to symmetric 3-stage Clos networks. All this work is done by Chung & Ross (Chung *et al.*, 1991), who have improved on the original work done by Melen & Turner (Melen *et al.*, 1989). Chung & Ross assumed that $\beta = 1$, that is, the input/output ports have the same capacity as the internal links. In the next subsections, we will extend their results to the case where the internal links can be faster than the input ports, (i.e., $\beta \leq 1$). We will also generalize these results for asymmetrical Clos networks.

Discrete Bandwidth case

Theorem 1 *Assuming each connection weight is a multiple of b and $1/b$ is integer, then a 3-stage Clos network is strictly nonblocking if and only if:*

$$M \geq 2 \cdot \left\lceil \frac{L - B}{1 - B + b} \right\rceil + 1 \quad (1)$$

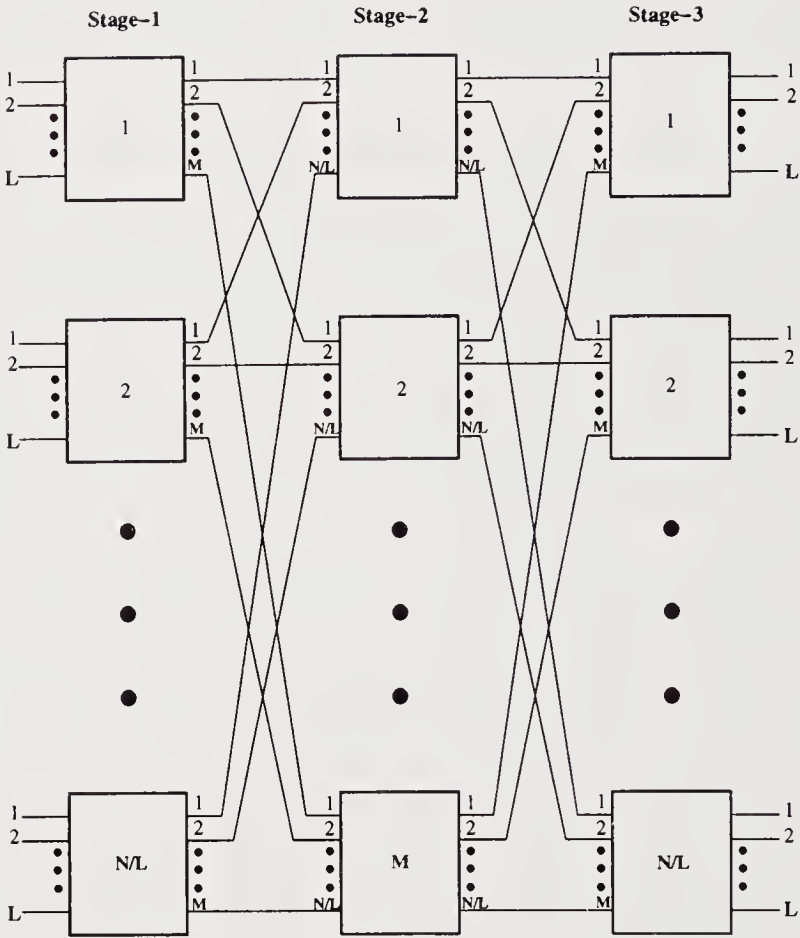


Figure 3 A symmetric three-stage Clos network.

Continuous Bandwidth case

Theorem 2 For $b > 0$ and $B \in (1 - b, 1]$, a 3-stage Clos network is strictly nonblocking if and only if:

$$M \geq 2 \cdot \left\lceil \frac{1}{b} \right\rceil \cdot (L - 1) + 1 \quad (2)$$

Theorem 3 For $b = 0$ and $B \in (0, 1)$, a 3-stage Clos network is strictly nonblocking if and only if:

$$\begin{aligned} M \geq M' &\stackrel{\text{def}}{=} \lim_{\epsilon \rightarrow 0^+} 2 \cdot \left\lceil \frac{L - 1}{1 - B + \epsilon} \right\rceil + 1 \\ &= \begin{cases} 2 \cdot \left\lceil \frac{L-1}{1-B} \right\rceil + 1 & \text{if } \frac{L-B}{1-B} \text{ not integer,} \\ 2 \cdot \left\lceil \frac{L-1}{1-B} \right\rceil - 1 & \text{if } \frac{L-B}{1-B} \text{ is integer.} \end{cases} \end{aligned}$$

Theorem 4 For $B = k \cdot b$ and $k, 1/b$ integers,

$$2 \cdot \left\lfloor \frac{L - B}{1 - B + b} \right\rfloor + 1 \leq M^* \leq 2 \cdot \left\lfloor \frac{L - B}{1 - B} \right\rfloor + 1 \quad (3)$$

In the following theorems, we will extend the above conditions by assuming that the internal links have a higher capacity than the input/output ports. In other words, we incorporate the *speedup factor* of the internal links into the already existing formulas by assuming that the normalized i/o port capacity, β , is in general less than or equal to the internal link capacity (which is normalized to 1).

At this point, we observe that there is no tight bound for the continuous bandwidth case, when $B \in [0, 1 - b]$. This case has been an open problem so far (Chung *et al.*, 1991) and, in this paper, we provide a solution for it in Theorem 9.

3.2 Extensions for more General Multirate Clos Networks

Assuming that the maximum i/o port capacity is β times the internal link capacity, where $0 \geq b \geq B \geq \beta \geq 1$, we can normalize with respect to the i/o port capacity and apply into the formulas derived by Chung and Ross:

$$0 \geq \frac{b}{\beta} \geq \frac{B}{\beta} \geq 1 \geq \frac{1}{\beta} \quad \longrightarrow \quad 0 \geq b' \geq B' \geq 1$$

Discrete Bandwidth case

Theorem 5 Assuming each connection weight is a multiple of b and $1/b$ is an integer, then a 3-stage Clos network is strictly nonblocking if and only if:

$$M \geq 2 \cdot \left\lfloor \frac{\beta L - B}{1 - B + b} \right\rfloor + 1 \quad (4)$$

when $\beta L \geq 1 + b$. Otherwise, $M \geq 1$.

Proof of Theorem 5

Sufficiency

Assume an incoming connection of weight ω . Since $1/b$ and ω/b are integers, the minimum capacity of an internal link that would block this connection would be $1 - \omega + b$. The maximum utilized bandwidth on the input links that would still allow the incoming connection to be established is $\beta L - \omega$. Therefore, the number of internal links that can be saturated with this load is at most $f(\omega) = \left\lfloor \frac{\beta L - \omega}{1 - \omega + b} \right\rfloor$. For $\beta L \geq 1 + b$, $f(\omega)$ is an increasing function of ω and, therefore, is maximized for $\omega = B$ in the interval $[b, B]$. For $\beta L < 1 + b$, $f(\omega)$ is a decreasing function of ω and, therefore, is maximized for $\omega = b$ in the interval $[b, B]$. With similar reasoning, the same number of middle-stage switches can be blocked by a third-stage switch. Hence we need at least twice as many middle-stage switches plus one, to ensure nonblocking operation in the worst case. ■

Necessity

Consider a configuration with $2(\beta L - B)/b$ connections, each with weight b . Half of them use the same first-stage switch u , and the same third-stage switch z , and they contribute

a weight of at least $1 - B + b$ to each of $\lfloor (\beta L - B)/(1 - B + b) \rfloor$ middle-stage switches. The remaining half of the connections, use in a similar way, first-stage switch $w \neq u$ and third-stage switch $v \neq z$. The two sets of middle-stage switches can be disjoint in the worst case. An incoming connection (u, v, B) would then require at least one more middle-stage switch to get through.

In the case where $\beta L < 1 + b$, the entire input capacity of a first-stage switch can fit in exactly one internal link. Hence, $M=1$. ■

Continuous Bandwidth case

Theorem 6 For $b > 0$ and $B \in (1 - b, \beta]$, a 3-stage Clos network is strictly nonblocking if and only if:

$$M \geq 2 \cdot \left\lfloor \frac{\beta}{b} \right\rfloor \cdot (L - 1) + 1 \quad (5)$$

Proof of Theorem 6

Sufficiency

Assume a 3-stage Clos network with a single first-stage and third-stage switch with L input ports. Without loss of generality, assume that the first input port has a weight $\leq 1 - \omega$, that is, it will allow an incoming connection with weight ω . Let $J(\omega, L)$ be the maximum number of internal links that have a weight $> 1 - \omega$. First, we will show that $J(\omega, L) \leq \lfloor \beta/b \rfloor (L - 1)$.

Let R be a configuration of connections. For each connection $r \in R$, let a_r be its weight. Let G_l be the set of connections in R that pass through the l^{th} input port. Thus, $\{G_1, \dots, G_L\}$ is a partition of R . Let, also, \mathcal{J} be the set of all internal links with weight $> 1 - \omega$. We observe that $J = |\mathcal{J}|$. Let H_j be the set of connections in R that pass through the j^{th} link in \mathcal{J} . Then:

$$\sum_{r \in G_1} a_r \leq \beta - \omega \quad (6)$$

$$\sum_{r \in G_l} a_r \leq \beta, \quad l = 2, \dots, L \quad (7)$$

$$\sum_{r \in H_j} a_r > 1 - \omega, \quad j \in \mathcal{J} \quad (8)$$

$$a_r \in [b, B], \quad r \in R \quad (9)$$

Let $G \stackrel{\text{def}}{=} \bigcup_{l=2}^L G_l$. Then,

$$|H_j \cap G| \geq 1, \quad j \in \mathcal{J} \quad (10)$$

otherwise, $\exists j \in \mathcal{J}$, such that $H_j \subseteq G_1 \Rightarrow \sum_{r \in H_j} a_r \leq \sum_{r \in G_1} a_r \leq \beta - \omega$, which contradicts (8), since $\beta \leq 1$. From (10) we have,

$$J \leq \sum_{j \in \mathcal{J}} |H_j \cap G| \leq |G| \quad (11)$$

From (7) and (9) we have $|G_l| \leq \lfloor \beta/b \rfloor, l = 2, \dots, L$. For $\omega = B$, (6) implies $|G_1| \leq \lfloor (\beta - B)/b \rfloor$. Hence,

$$|G| = \sum_{l=2}^L G_l \leq \left\lfloor \frac{\beta}{b} \right\rfloor (L-1). \quad (12)$$

From (11) and (12), we conclude that $J(\omega, L) \leq \lfloor \beta/b \rfloor (L-1)$. Therefore, we need at least one more than twice as many middle-stage switches for nonblocking operation. ■

Necessity

For every $B \in (1-b, \beta]$, a single connection (even of weight b) is enough to block an internal link, since in the worst case, $b + B > 1$. Consider a configuration consisting of $\lfloor \beta/b \rfloor (L-1)$ connections of weight b , with exactly one connection saturating each internal link. Then, a new connection of weight B will be blocked unless we have at least $2\lfloor \beta/b \rfloor (L-1) + 1$ middle-stage switches. ■

Theorem 7 For $b = 0$ and $B \in (0, \beta), \beta < 1$, a 3-stage Clos network is strictly nonblocking if and only if:

$$M \geq 2 \cdot \left\lceil \frac{\beta L - 1}{1 - B} \right\rceil + 1 = 2 \cdot \left\lceil \frac{\beta L - B}{1 - B} \right\rceil - 1 \quad (13)$$

Proof of Theorem 7

Let M^* be the minimum number of middle-stage switches for the 3-stage Clos network to be strictly nonblocking. We observe that the case of the continuous bandwidth for $b = 0$ can be solved by considering the discrete bandwidth case with arbitrarily small b . Therefore, from theorem 5 and the observation that

$$\lim_{b \rightarrow 0^+} \left\lceil \frac{\beta L - B}{1 - B - b} \right\rceil = \left\lceil \frac{\beta L - B}{1 - B} \right\rceil - 1$$

we have

$$M^* = 2 \cdot \lim_{b \rightarrow 0^+} \left\lceil \frac{\beta L - B}{1 - B - b} \right\rceil + 1 = 2 \cdot \left\lceil \frac{\beta L - B}{1 - B} \right\rceil - 1 = 2 \cdot \left\lceil \frac{\beta L - 1}{1 - B} \right\rceil + 1. \quad (14)$$

■

Theorem 8 For $B = k \cdot b$ and $k, \beta/b$ integers,

$$2 \cdot \left\lceil \frac{\beta L - B}{1 - B + b} \right\rceil + 1 \leq M^* \leq 2 \cdot \left\lceil \frac{\beta L - B}{1 - B} \right\rceil - 1 \quad (15)$$

Proof of Theorem 8

The lower bound is derived from theorem 5, if we consider the discrete bandwidth case as a sub-case of the continuous bandwidth case (*necessary condition*).

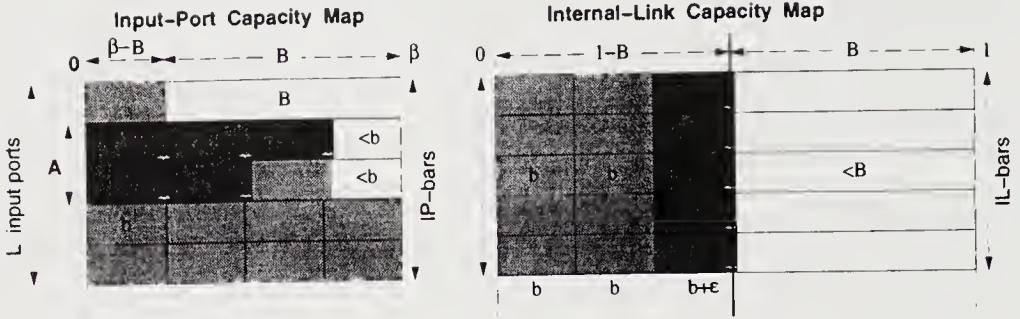


Figure 4 Illustration to explain the proof of Theorem 9 when $1/b$, B/b and β/b are integers.

The upper bound is derived from the observation that every configuration of connections with $\omega \in [b > 0, B]$ is also a configuration of connections with $\omega \in [0, B]$. Therefore, the upper bound is the exact bound of theorem 7 (*sufficient condition*). ■

The following theorem gives the necessary and sufficient condition for strictly nonblocking operation of 3-stage Clos networks, in the continuous-bandwidth multirate environment, for a special case which has been identified as an open research problem by Chung & Ross (Chung *et al.*, 1991).

Theorem 9 Assuming that $b > 0$, $B < 1$, and either $B > 1 - b$ or all $1/b$, B/b and β/b are integers, then a 3-stage Clos network is strictly nonblocking in the continuous-bandwidth multirate environment if and only if

$$M \geq 2 \cdot \varphi + 1 = 2 \cdot \left\lfloor \frac{(L-1) \left\lfloor \frac{\beta}{b} \right\rfloor + \left\lfloor \frac{\beta-B}{b} \right\rfloor - A}{\left\lfloor \frac{1-B}{b} \right\rfloor} \right\rfloor + 1 \quad (16)$$

where

$$A = \begin{cases} 0, & \text{if } B > 1 - b, \\ \left\lfloor \frac{b \cdot \varphi}{\beta - b} \right\rfloor, & \text{otherwise} \end{cases} \quad (17)$$

Proof of Theorem 9

Let us first consider the case when $B > 1 - b$. This implies

$$\beta - B \leq 1 - B \leq b$$

and the theorem becomes equivalent to theorem 6.

Let us now consider the case when all $1/b$, B/b and β/b are integers (refer to Figure 4).

First, we will show a constructed configuration that achieves the bound suggested by the theorem (*necessity*), and then we will show that any other configuration cannot exceed this bound (*sufficiency*). Therefore, the bound is exact.

Necessity:

Consider a configuration such that all connections have weights either b or $b + \epsilon$, where ϵ is some arbitrarily small positive number. Let us call these connections **SMALLBLOCK** and **BIGBLOCK**, respectively.

First, we saturate a number φ of internal links (IL-bars) by routing through each of them one **BIGBLOCK** and $(\frac{1-B-b}{b})$ **SMALLBLOCK**(s). Then, we “tile” the input port capacity map as shown in Figure 4. Due to the fact that an integer number of **BIGBLOCK**s does not fit exactly in an IP-bar, there is some unused capacity in the input ports, (besides the space allocated on the first input port for the next connection). Since we loose exactly one **SMALLBLOCK** for each IP-bar that contains at least one **BIGBLOCK**, the total number of **SMALLBLOCK**s that we lose from a perfect tiling is given by

$$A = \left\lceil \frac{\varphi}{\frac{\beta}{b} - 1} \right\rceil = \left\lceil \frac{b\varphi}{\beta - b} \right\rceil \quad (18)$$

The total capacity used in the input ports (i.e., the tiled area) is equal to the total capacity used in the internal links. Therefore, the maximum number of saturated links is:

$$\varphi = \left\lfloor \frac{\beta L - B - bA}{1 - B} \right\rfloor = \left\lfloor \frac{(L - 1) \left\lfloor \frac{\beta}{b} \right\rfloor + \left\lfloor \frac{\beta - B}{b} \right\rfloor - A}{\left\lceil \frac{1 - B}{b} \right\rceil} \right\rfloor \quad (19)$$

Hence, the condition $M \geq 2\varphi + 1$ is necessary.

Sufficiency:

Let us now consider some configuration \mathcal{R} of arbitrary-weight connections, with total used input capacity U_R . Let also U_C be the total used input capacity of the constructed configuration, \mathcal{C} , above. Then, $U_C = (\beta L - B - bA)/(1 - B)$.

Obviously, if $U_R \leq U_C$, then \mathcal{R} does not require more than $2\varphi + 1$ middle-stage switches for strictly nonblocking operation.

If $U_R > U_C$, then the only input ports that can be further loaded in \mathcal{R} are the ones that contain at least one **BIGBLOCK** in \mathcal{C} . Therefore, the difference $U_R - U_C > 0$ can be distributed (in the worst case) by simply augmenting the **BIGBLOCK**s by an extra weight $w \in [0, \min\{b, B - b\})$. Since all saturated links in \mathcal{C} can augment their **BIGBLOCK**s by any weight less than B , we conclude that \mathcal{R} will not require more middle-stage switches than \mathcal{C} for strictly nonblocking operation. ■

One method to solve the co-dependent equations (16) and (17) for φ is as follows:

1. Set $A = 0$ in (16) and compute an upper bound for φ .
2. Compute A from (17).
3. If both equations are not satisfied by the computed values of A and φ let $\varphi := \varphi - 1$ and goto step 2.

The above method eventually converges to the solution within a small number of iterations, since (i) a solution always exists by construction (see proof), and (ii) the value of A is usually small.

In the special case, where all $1/b$, B/b and β/b are integers, Equation (16) of Theorem 9 becomes

$$M \geq 2 \cdot \varphi + 1 = 2 \cdot \left\lfloor \frac{L \cdot \beta - B - b \cdot A}{1 - B} \right\rfloor + 1 \quad (20)$$

The following example, presented in Chung & Ross (Chung *et al.*, 1991), illustrates such a case.

Example 1 Consider a symmetric 3-stage Clos network with $L = 5$ links per input switch, $b = 0.1$, $B = 0.8$ and $\beta = 1$. For this case, Theorem 8 bounds the minimum number of middle-stage switches required for strictly nonblocking operation between 29 and 41. Theorem 9 gives $M^* = 39$, which verifies the result reported by Chung & Ross (Chung *et al.*, 1991). ■

Corollary 1 For every $b > 0$, $B < 1$, the bound established in Theorem 9 is a lower bound for the minimum number of middle-stage switches required for strictly nonblocking operation.

The proof follows from the fact that the cases of Theorem 9 are subcases of the corollary.

3.3 Asymmetrical Clos Networks

The following theorems extend the results presented in Section 3.2 for asymmetrical 3-stage Clos networks.

Let $R_i \stackrel{\text{def}}{=} \min_j R_{i,j}$ and $T_k \stackrel{\text{def}}{=} \min_j T_{j,k}$.

Discrete Bandwidth case

Theorem 10 Assuming each connection weight is a multiple of b and $R_i/b, T_k/b$ are integers, then an asymmetrical 3-stage Clos network is strictly nonblocking, if:

$$M \geq \max_i \left\lfloor \frac{\beta \cdot P_i - B}{R_i - B + b} \right\rfloor + \max_k \left\lfloor \frac{\beta \cdot Q_k - B}{T_k - B + b} \right\rfloor + 1 \quad (21)$$

when $\beta \cdot P_i \geq 1 + b$, and $\beta \cdot Q_k \geq 1 + b$. Otherwise, $M \geq 1$.

Proof of Theorem 10

In the more general case of an asymmetrical Clos network, each first-stage switch i has P_i input ports and each third-stage switch k has Q_k output ports. In this case, the condition of Theorem 5 must hold for *every* first-stage and third-stage switch. Therefore, we can apply a similar proof as in Theorem 5. Let $L_1 = \max\{P_i\}$ for the first-stage switches and $L_2 = \max\{Q_k\}$ for the third-stage switches. Now, assume a symmetric switch consisting of the worst case combination of one input switch with L_1 input ports, one output switch with L_2 output ports and internal links with capacities R_i and T_k between the three stages, respectively. To simplify our results, we used $R_i \stackrel{\text{def}}{=} \min_j R_{i,j}$ and $T_k \stackrel{\text{def}}{=} \min_j T_{j,k}$ in the denominator. This simplifying assumption yields the sufficient condition given in the theorem. ■

Continuous Bandwidth case

Theorem 11 For $b > 0$ and $B \in (1 - b, \beta]$, an asymmetrical 3-stage Clos network is strictly nonblocking, if:

$$M \geq \left\lfloor \frac{\beta}{b} \right\rfloor \cdot (\max_i \{P_i\} + \max_k \{Q_k\} - 2) + 1 \quad (22)$$

Theorem 12 For $b = 0$ and $B \in (0, \beta)$, an asymmetrical 3-stage Clos network is strictly nonblocking, if:

$$M \geq \max_i \left\lceil \frac{\beta \cdot P_i - B}{R_i - B} \right\rceil + \max_k \left\lceil \frac{\beta \cdot Q_k - B}{T_k - B} \right\rceil - 1 \quad (23)$$

Theorem 13 For $B = k \cdot b$ and $k, \beta/b$ integers, in an asymmetrical 3-stage Clos network:

$$\begin{aligned} \max_i \left\lceil \frac{\beta \cdot P_i - B}{R_i - B + b} \right\rceil + \max_k \left\lceil \frac{\beta \cdot Q_k - B}{T_k - B + b} \right\rceil + 1 \leq M^* \leq \\ \max_i \left\lceil \frac{\beta \cdot P_i - B}{R_i - B} \right\rceil + \max_k \left\lceil \frac{\beta \cdot Q_k - B}{T_k - B} \right\rceil - 1 \end{aligned} \quad (24)$$

The proofs of Theorems 11-13 are similar to the proofs of Theorems 6-8 in combination with the observations of the proof of Theorem 10. ■

4 CONCLUDING REMARKS

In this paper we studied the strictly nonblocking switching operation of 3-stage Clos networks in the multirate environment.

First, we generalized the results of Table 1.4 for the case, when the internal links have higher bandwidth capabilities than the input/output ports, by introducing the internal-link speedup factor $(1/\beta)$. Subsequently, we extended these results for general asymmetrical Clos networks. In addition to these extensions, we contributed a general tight bound for the case of multirate traffic, where $1/b, B/b, \beta/b$ are all integers, and B is independent of b . This formula also generates earlier reported bounds as special cases.

REFERENCES

- Beneš, V.E. (1965) *Mathematical Theory of Connecting Networks and Telephone Traffic*. Academic Press, New York.
- Cantor, D.G. (1971) On Non-Blocking Switching Networks. *Networks*, **1**, 367-377.
- Clos, C. (1953) A study of non-blocking switching networks. *Bell Systems Technical Journal*, 406-424.
- Chung, S.-P. and Ross, K.W. (1991) On Nonblocking Multirate Interconnection Networks. *SIAM Journal on Computing*, (**20,4**), 726-736.

- Collier, M. and Curran, T. (1994) The strictly non-blocking condition for three-stage networks. *Proc. of the 14th International Teletraffic Congress - ITC 14*, 635-644, Antibes Juan-les-Pins, France.
- Coudreuse, J.P. and Servel, M. (1987) Prelude: An Asynchronous, Time-Division Switched Network. *International Communications Conference*.
- Feldman, P. Friedman, J. and Pippenger, N. (1986) Non-Blocking Networks. *Proceedings of STOC 1986*, 247-254.
- Huang, A. and Knauer, S. (1984) Starlite: a Wideband Digital Switch. *Proceedings of Globecom-84*, 121-125.
- Jajszczyk, A. (1983) On nonblocking switching networks composed of digital symmetrical matrices. *IEEE Transactions on Communications*, **COM-31**, 2-9.
- Masson, G.M. Gingher, G.C. and Nakamura, S. (1979) A Sampler of Circuit Switching Networks. *Computer*, 145-161.
- Melen, R. and Turner, J.S. (1989) Nonblocking Multirate Networks. *SIAM Journal on Computing*, (**18,2**), 301-313.
- Pippenger, N. (1982) Telephone Switching Networks. *Proceedings of Symposia in Applied Mathematics*, **26**, 101-133.
- Turner, J.S. (1988) Design of a Broadcast Packet Network. *IEEE Transactions on Communications*, 734-743.
- Varma, A. and Chalasani, S. (1993) Asymmetrical Multiconnection Three-Stage Clos Networks. *Networks*, **23**, 427-439.

5 BIOGRAPHIES

Fotios K. Liotopoulos (IEEE '91 / ACM '91) was born in Kozani, Greece, in 1965. He received the B.Tech. degree in Computer Engineering and Information Sciences from the University of Patras, Greece, in 1988, the M.Sc. degree in Computer Sciences, in 1991, and the Ph.D. degree in Electrical and Computer Engineering, in 1996, both from the University of Wisconsin-Madison. His research interests include computer interconnection networks, parallel architectures and algorithms and fast switch architectures and performance. Dr. Liotopoulos is a member of the Sigma-Xi Scientific Research Society and the Eta-Kappa-Nu Engineering Society. His e-mail address is: fotios@ece.wisc.edu

Suresh Chalasani received the B.Tech. degree in Electronics and Communication Engineering from the JNT University, Hyderabad, India, in 1984, the M.E. degree in Automation from the Indian Institute of Science, Bangalore, India, in 1986, and the Ph.D. degree in Computer Engineering from the University of Southern California in 1991. He is currently an assistant professor of Electrical and Computer Engineering at the University of Wisconsin-Madison. His research interests include parallel architectures, parallel algorithms and fault-tolerant systems. Prof. Chalasani's e-mail address is: suresh@ece.wisc.edu

Performance Analysis of Buffered Banyan ATM Switch Architectures^{*}

D. Kouvatsos, J. Wilkinson,

Computer Systems Modelling Group, Dept. of Computing, University of Bradford, BRADFORD BD7 1DP, U.K.

Tel: 01274 383941 Fax: 01274 383920

Email: D.D.Kouvatsos, J.Wilkinson @comp.brad.ac.uk,

P. Harrison, M. Bhabuta

Dept. of Computing, Imperial College of Science, Technology & Medicine, 180 Queens Gate, LONDON SW7 2BZ, U.K.

Tel: 0171 5948363 Fax: 0171 5818024

Email: pgh,mb3@doc.ic.ac.uk

Abstract

The principle of Maximum Entropy (ME) and the notion of system decomposition are combined towards the creation of an iterative cost-effective approximation algorithm for the performance analysis of packet-switched buffered Banyan Multistage Interconnection Network (MIN) based Asynchronous Transfer Mode (ATM) switch architectures with arbitrary buffer sizes, multiple input/output ports and Repetitive Service (RS) internal blocking.

Traffic entering and flowing in the MIN is assumed to be bursty and it is modelled by a Compound Poisson Process (CPP) with geometrically distributed bulk sizes and Generalised Exponential (GE) interarrival times. The GE distribution is also adopted to represent the random nature of the effective service times of packets due to the combined effects of traffic burstiness and RS blocking.

Entropy maximisation implies decomposition of the Banyan network into individual building block queues of switching elements, represented by shared buffer cross bars, under revised GE-type interarrival and service times. Each building block queue is analysed in isolation by applying ME techniques and classical queueing theory, subject to marginal mean value constraints, in order to obtain a product form solution for the joint queue length distribution and typical performance metrics of the network.

Numerical results are included to validate the credibility of the ME approximation against simulation, define experimental performance bounds and perform a buffer capacity optimisation across the entire network.

^{*} Supported by the Engineering and Physical Sciences Research Council (EPSRC), UK, under grant GR/K/67809.

Keywords

Multistage Interconnection Network (MIN), Banyan network, Queueing Network Model (QNM), Repetitive-Service (RS) blocking mechanism, Maximum Entropy (ME) Principle, Compound Poisson Process (CPP), Generalised Exponential (GE) distribution, Asynchronous Transfer Mode(ATM) switch architectures.

1 INTRODUCTION

During the past decade, a considerable amount of effort has been made towards the design and development of Asynchronous Transfer Mode (ATM) switch architectures, which are widely considered as the preferred packet-oriented solution of a new generation of high speed communication systems, both for broadband public information highways and for local and wide area private networks (e.g., Tobagi [25]).

Amongst the many types of ATM switch architectures, of particular interest are the so called space division switches which are primarily based on Multistage Interconnection Networks (MINs) (e.g., [1,2,19]). Such switches are composed of smaller switching elements represented by shared-buffer crossbars. Main features of a MIN include non-centralised switching control and multiple concurrent paths in tandem from input ports to output ports.

MINs are also widely employed in parallel processing systems as a means for processor - memory (and interprocessor) communication. The nature of traffic in ATM switches, however, is quite different from that observed in typical parallel machines in the sense that, regarding the latter, there is basically only one type of service, namely, high speed data (not considering "probe" and "acknowledgment" signals observed in inter-stage transmissions), whereas for the former, there exists a greater variety of integrated services including voice, low and high speed data, teleconferencing, TV distribution and video on demand, all of which share the same communication medium with different cell loss and delay requirements.

The integration of such ATM services implies considerable variability in terms of transmission speed and holding times. Moreover, the flow of cells through one switching element may be momentarily blocked (halted) if the downstream switching element has reached its buffer capacity. Thus, credible analytical tools are essential for the cost-effective performance modelling prediction of such complex ATM switches.

An increasing number of earlier papers concerning with the performance modelling and analysis of MINs have appeared in the literature (e.g., [4-6, 8, 20, 24, 26]) and such trend is likely to continue towards the design and development of more appropriate ATM space division architectures. In this context, analytic performance models of shared buffer ATM switch architectures, based on both continuous-time and discrete-time queueing models, have received particular attention. Pinto and Harrison [4, 5] proposed approximate algorithms for the analysis continuous-time asynchronous buffered Banyan networks with 2x2 switching elements using Exponential interarrival times and 2-phase Coxian (C_2) and Generalised Exponential (GE) service time distributions, respectively, with Blocking After Service (BAS) (i.e., service is suspended at the output port for a cell which attempts to enter a destination switching element with a full buffer). Hong et al [6] and Yamashita et al [26] described approximate algorithms for the performance evaluation of discrete-time and continuous-time queueing models of shared buffer ATM switches under both Interrupted Bernoulli and Interrupted Poisson arrival processes, respectively. In terms of computational implementation,

these works tackle the problem by either solving global balance equations numerically [4, 5], or by decomposing the switch into several subsystems, each of which being analysed numerically in isolation [6, 26]. However, as the number of input (or output) ports increases, so does the size of the system's state space, and therefore, further approximations are required in order to achieve, if at all possible, tractable solutions. Thus, there is a great need to apply alternative methodologies leading to both accurate and cost-effective approximations for the performance modelling and evaluation of MIN-based shared buffer ATM switches.

The principle of Maximum Entropy (ME), a probability inference method (c.f., Jaynes [7], Shore and Johnson [22]), has been used successfully, in conjunction with queueing theoretic mean value constraints, for the approximate analysis of both continuous time and discrete time arbitrary Queueing Network Models (QNMs) with single general queues of finite or infinite capacity (e.g., [10-17]). In particular, the principle has been utilised in the study of general multibuffered and shared buffer queues and closed form expressions in both continuous-time and discrete-time domains have been obtained for Queue Length Distributions (QLD), Cell Loss Probabilities (CLP) and mean delays [14, 15]. More recently, a new product from approximation has been established by Kouvatso and Wilkinson [17], towards the cost-effective performance analysis of arbitrary open discrete-time QNMs of shared buffer queues with cell loss. In the aforementioned studies the arrival process at each queue has been assumed to be highly variable and was modelled by Compound Poisson (CPP) or Bernoulli (CBP) processes, both with geometrically distributed bulk sizes. In this context, the burstiness of the arrival process is characterised by the squared coefficient of variation (SCV) of the interarrival times or, equivalently, the average size of the incoming bulk. The CPP and CBP arrival processes imply GE and Generalised Geometric (GGeo) interarrival-time distributions, respectively, whose pseudo-memoryless properties facilitate the analysis of complex queues and networks (e.g., [11, 13, 16]). The choice of GE and GGeo distributions has been further motivated by the fact that measurements of actual traffic or service times are generally limited and so only few parameters can be computed reliably. Typically, only the mean and variance can be relied upon. In this case, the choice of distributions which imply least bias (c.f., [7]) (i.e., introduction of arbitrary and, therefore, false assumptions) is that of a GE or GGeo distribution within a continuous-time or a discrete time context, respectively.

In this paper queueing network modelling and entropy maximisation are employed towards the performance analysis of Banyan MINs with GE-type external traffic pattern and stage-to-stage transmission times, arbitrary switching element sizes ($R \times R$, $R \geq 2$) and buffer capacities, K , under Repetitive-Service (RS) (or communication) internal blocking. Such MINs provide full connectivity between a set of input sources and a set of destination nodes. In a Broadband Integrated Services Digital Network (B-ISDN) environment, Banyan MINs can support several different types of traffic concurrently (e.g., data, voice, video). Consequently, traffic models must be able to capture various flow characteristics such as burstiness (e.g. video traffic which has to be batched). In this context, the GE distribution is adopted to represent (in an appropriate fashion) the random nature of the interarrival times and effective service times of packets in the MIN due to the combined influence of traffic burstiness and RS blocking. Note that in tandem configurations RS blocking occurs when a cell upon service completion at queue κ attempts to join a downstream queue ℓ whose buffer capacity is full. Consequently, the cell is rejected by queue ℓ and immediately receives another service at queue κ . This is

repeated until the cell completes service at queue κ at the moment where the destination queue ℓ is not full.

Entropy maximisation implies a decomposition of the Banyan network into individual multiple input GE-type shared buffer queues of switching elements with revised (effective) interarrival and transmission times. These queues are solved in isolation and together with GE-type formulae for the first two moments of the cell interdeparture and aggregated arrival processes at each output port queue, play the role of cost effective building blocks towards the performance analysis of the entire network.

The ME formalism is introduced in Section 2. The GE-type distribution is described in Section 3. An ME QLD of a multiple input shared buffer building block queue is outlined in Section 4. An ME product form approximation for a arbitrary QNM of a buffered Banyan MIN together with a description of the traffic flow through the switching elements are presented in Section 5. ME Analysis of three types of switching elements, acting as building blocks, together with appropriate GE flow formulae are presented in Section 6. Section 7 presents the ME approximation algorithm for the performance analysis of arbitrary size Banyan networks. Numerical results and concluding comments follow in Sections 8 and 9, respectively.

2 MAXIMUM ENTROPY FORMALISM

Consider a system Q which has a set S of possible discrete states $\{S_0, S_1, S_2, \dots\}$ which may be finite or countably infinite and state S_n , $n = 0, 1, 2, \dots$ may be specified arbitrarily. Suppose that the available information about Q places a number of constraints on $p(S_n)$, the probability distribution that the system Q is in state S_n . Without loss of generality, it is assumed that these constraints take the form of mean values of suitable functions $\{f_1(S_n), f_2(S_n), \dots, f_m(S_n)\}$, where m is less than the number of possible states. The principle of maximum entropy [7] states that, of all distributions which satisfy the constraints, the minimally biased distribution is the one which maximises the system's entropy function

$$H(p) = - \sum_{S_n \in S} p(S_n) \ln p(S_n), \quad (2.1)$$

subject to the constraints

$$\sum_{S_n \in S} p(S_n) = 1, \quad (2.2)$$

$$\sum_{S_n \in S} f_k(S_n) p(S_n) = \langle f_k \rangle, \quad k = 1, 2, \dots, m, \quad (2.3)$$

where $\{\langle f_k \rangle : k = 1, 2, \dots, m\}$ are the prescribed mean values defined on the set of m functions $\{f_k(S_n) : k = 1, 2, \dots, m\}$, where m is less than the number of states in S . The maximisation of

(2.1), subject to the constraints (2.2) and (2.3), can be carried out using Lagrange's method of undermined multipliers and leads to the solution

$$p(S_n) = \frac{1}{Z} \exp \left\{ - \sum_{k=1}^m \beta_k f_k(S_n) \right\}, \quad (2.4)$$

where $\{\beta_k : k = 1, 2, \dots, m\}$, are the Lagrangian multipliers determined from the set of constraints (2.3) and Z , known in statistical physics as the "partition function", is given by

$$Z = \exp(\beta_0) = \sum_{S_n \in S} \exp \left\{ - \sum_{k=1}^m \beta_k f_k(S_n) \right\}, \quad (2.5)$$

where $\{\beta_0\}$ is the Lagrangian multiplier determined by the normalisation constraint (2.2).

Jaynes [7] has shown that, if the prior information includes all constraints actually operative during a random experiment, the distribution predicted by the maximum entropy can be realised in overwhelmingly more ways than by any other distribution. The principle of maximum entropy has also been shown by Shore and Johnson [22] to provide a "uniquely correct self-consistent method of inference" for estimating probability distributions based on the available information.

Maximum entropy formalism can be applied in the performance analysis of queueing systems because expected values of various distributions of interest are usually known in terms of moments of the interarrival and service time distributions. A review of entropy maximisation for approximate analysis of queueing systems and networks can be seen in Kouvatsos [16].

3 THE GE DISTRIBUTION

The GE distribution is of the form

$$F(t) = P(X \leq t) = 1 - \tau e^{-\sigma t}, \quad t \geq 0 \quad (3.1)$$

where

$$\tau = 2 / (C^2 + 1), \quad \sigma = \tau \nu,$$

X is a mixed-time random variable (rv) of the interevent-time, while $1/\nu$ is the mean and C^2 is the SCV of rv X (see Figure 1).

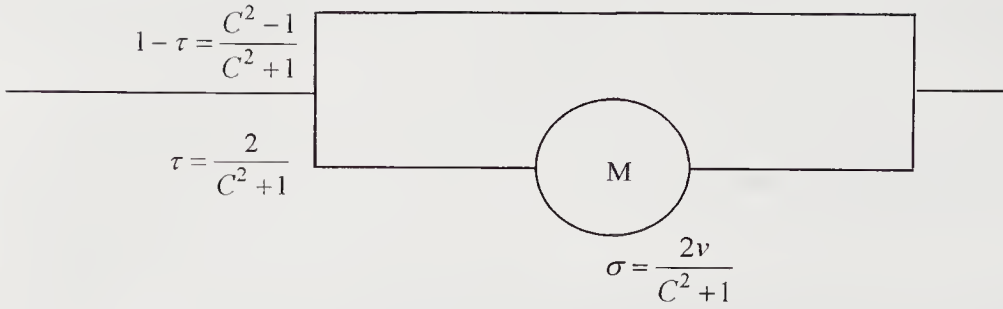


Figure 1 The $GE(\nu, C^2)$ distribution with parameters τ and σ .

For $C^2 \geq 1$, the GE model is a mixed-time probability distribution and it can be interpreted as either

1. an extremal case of the family of two-phase exponential (M) distributions (e.g., Hyperexponential-2 (H_2)) having the same ν and C^2 , where one of the two phases has zero service time, or
2. a bulk type distribution with an underlying counting process equivalent to a Compound Poisson Process (CPP) with parameter $2\nu / C^2 + 1$ and geometrically distributed bulk sizes with mean $= (C^2 + 1) / 2$ and $SCV = (C^2 - 1) / (C^2 + 1)$ given by

$$P(N_{cp} = n) = \begin{cases} \sum_{i=1}^n \frac{\sigma^i}{i!} e^{-\sigma} \binom{n-1}{i-1} \tau^i (1-\tau)^{n-i}, & \text{if } n \geq 1, \\ e^{-\sigma}, & \text{if } n = 0, \end{cases} \quad (3.2)$$

where N_{cp} is a Compound Poisson rv of the number of events per unit time corresponding to a stationary GE-type interevent rv.

The GE distribution is versatile, possessing pseudo-memoryless properties which make the solution of many GE-type queueing systems and networks analytically tractable (e.g., Kouvatso [16]). Moreover, it has been experimentally established that the GE model, due to its extremal nature, defines pessimistic performance bounds on typical performance measures over corresponding estimates based on two-phase distributions having the same first two moments as the GE. The GE distribution is completely characterised in terms of mean rate, ν and, SCV , C^2 and it can be interpreted as an ME solution (c.f., Jaynes [7]), subject to the constraints of normalisation, discrete-time zero probability and expected value. In this sense, it can be viewed as the least biased distribution estimate, given the available information in terms of the constraints.

For $C^2 < 1$, the GE distributional model (with $F(0) < 1$) cannot be physically interpreted as a stochastic model. However, it can be meaningfully considered as a pseudo-distribution function of a flow model approximation of an underlying stochastic model in which negative branching pseudo-probabilities (or weights) are permitted. To this end, all analytical GE-type

exact and approximate results obtained for queueing systems and networks when $C^2 < 1$ can also be used - by analogy - as useful heuristic approximations when $C^2 < 1$ as long as they satisfy basic queueing theoretic constraints (c.f. [16]). Note that utility of other improper two-phase type distributions (with $C^2 < 1$) in the field of systems modelling has been proposed by various authors (e.g., Nojo and Watanabe [21], Sauer [23]).

4 ME ANALYSIS OF A SHARED BUFFER QUEUE

Consider a general queueing model of a shared buffer switching element with bursty arrivals, depicted in Figure 2. The queueing model consists of R parallel single server queues, where R is the number of output ports. Each server represents an output port and each queue corresponds to the address queue for the output port. There are $R \times R$ bursty and heterogeneous GE-type interarrival streams of cells, R (multiple) streams to each of R input ports. Each stream has a mean overall arrival rate, Λ_{ji} , of cells and a SCV of interarrival time, Ca_{ji}^2 , for stream (j,i) , $i,j=1,2,\dots,R$ (n.b., subscript j is dropped in the case of a single stream per input port). Similarly, the transmission (or service) time of a cell at queue i follows a GE distribution with mean rate μ_i , and SCV Cs_i^2 , for stream i , $i=1,2,\dots,R$. Let K be the size of the total shared buffer. A cell is lost if it arrives at a time when there is a total of K cells in the R queues. Without loss of generality, it is assumed that any of the R queues may attain the maximum size K .

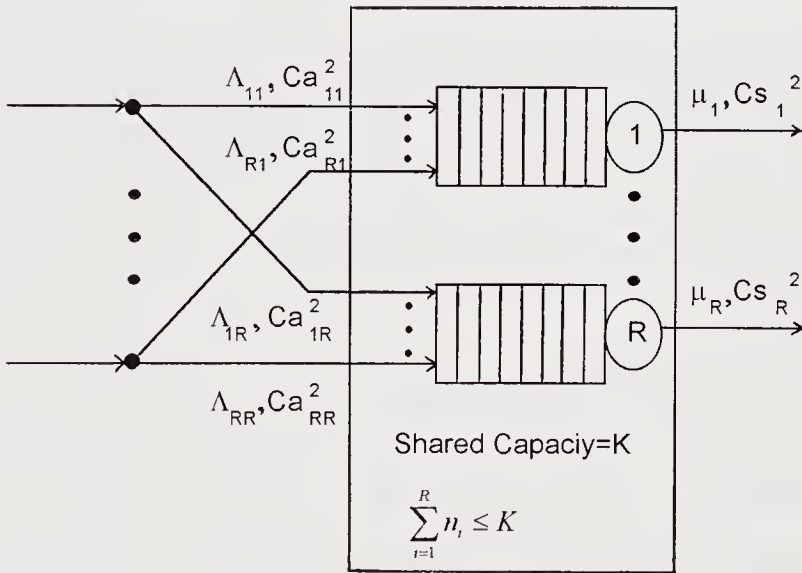


Figure 2 The $S_{RXR}(GE^R / GE / 1 / K)$ queueing model of a shared buffer switch.

The queueing model of the shared buffer switching element is denoted by $S_{R \times R}(GE^R / GE / 1 / K)$, such that

1. The overall interarrival times and service times at an $R \times R$ shared buffer queue are heterogeneous and GE distributed,
2. Each output port has a single server,
3. The total shared buffer capacity of the switch is K .

Moreover, let the state of the system at any given time be represented by a vector $\mathbf{n} = (n_1, n_2, \dots, n_R)$, where n_i is the number of cells in queue $i, i=1, 2, \dots, R$, and

$$\mathbf{n} \in S(K, R) = \left\{ \mathbf{n} = (n_1, n_2, \dots, n_R) : \sum_{i=1}^R n_i \leq K, 0 \leq n_i \leq K, i = 1, \dots, R \right\}.$$

Also let $p(\mathbf{n}), \mathbf{n} \in S(K, R)$, be the joint state probability distribution.

Note that the ME solution of the $S_{R \times R}(GE^R / GE / 1) / K$ queueing system, $p(\mathbf{n})$ is of the same form as the ME solution of an $S_{R \times R}(GE / GE / 1) / K$ queueing system with a single (merged) arrival stream at each of the R input points (c.f. Kouvatso[14]), subject to a common set of mean value constraints. Both solutions are presented below.

4.1 An ME Solution for the $S_{R \times R}(GE / GE / 1) / K$ Queueing System: an Outline

The form of the ME solution of an $S_{R \times R}(G / G / 1) / K$ queueing system, subject to normalisation and the constraints: server utilisation, $U_i, 0 < U_i < 1$; MQL $L_i, U_i \leq L_i < K$; conditional aggregate probability ϕ_i of a full buffer subject to $0 < \phi_i < 1, n_i > 0, i = 1, 2, \dots, R$, is given by the method of Lagrange's undetermined multipliers as (c.f. (2.4))

$$p(\mathbf{n}) = \frac{1}{Z} \prod_{i=1}^R g_i^{s_i(\mathbf{n})} x_i^{n_i} y_i^{f_i(\mathbf{n})}, \quad \forall \mathbf{n} \in S(K, R), \quad (4.1)$$

where Z is the normalising constant

$$Z = \sum_{\mathbf{n} \in S(K, R)} \prod_{i=1}^R g_i^{s_i(\mathbf{n})} x_i^{n_i} y_i^{f_i(\mathbf{n})},$$

$s_i(\mathbf{n})$ and $f_i(\mathbf{n})$ are auxiliary (indicator) functions defined by

$$s_i(\mathbf{n}) = \begin{cases} 1, & n_i > 0, \\ 0, & \text{otherwise,} \end{cases}$$

$$f_i(\mathbf{n}) = \begin{cases} 1, & \sum_{j=1}^R n_j = K \wedge s_i(\mathbf{n}) = 1, \\ 0, & \text{otherwise,} \end{cases}$$

and $\{g_i, x_i, y_i: i = 1, 2, \dots, R\}$ are the GE-type Lagrangian coefficients corresponding to the constraints $\{U_i, L_i, \varphi_i: i = 1, 2, \dots, R\}$, respectively.

Lagrangian coefficients $\{g_i, x_i: i = 1, 2, \dots, R\}$ are obtained by making asymptotic connections with the ME solution of a stable GE/GE/1 queue (c.f., [11]), namely

$$g_i = \frac{\rho_i(1-x_i)}{x_i(1-\rho_i)}, \quad x_i = \frac{L_i - \rho_i}{L_i}, \quad \rho_i = \Lambda_i / \mu_i, \quad i = 1, 2, \dots, R, \quad (4.2)$$

$$\text{where } L_i = \frac{\rho_i}{2} \left(1 + \frac{Ca_i^2 + \rho_i^2 Cs_i^2}{1 - \rho_i} \right), \quad i = 1, 2, \dots, R,$$

(i.e., g_i and x_i are assumed to be invariant to the buffer size K).

Moreover, Lagrangian coefficients $\{y_i: i = 1, 2, \dots, R\}$ can be computed by

1. Focusing on the flow balance equations

$$\Lambda_i(1 - \pi_i) = \mu_i U_i, \quad i = 1, 2, \dots, R, \quad (4.3)$$

where π_i is the cell loss probability for an attempted arrival to the output port queue i ,

2. Deriving recursive expressions for π_i and U_i , $i = 1, 2, \dots, R$, and
3. Solving numerically the resultant non-linear simultaneous equations, (n.b., for $R=2$, these equations can be solved analytically - see formulae (4.18)).

The normalising constant can be determined by applying the generating function approach and can be computed recursively by [14]

$$Z = \sum_{v=0}^{K-1} C_1(v) + C_2(K), \quad (4.4)$$

where $\{C_1(v): v = 0, 1, \dots, K-1\}$ and $\{C_2(K)\}$ are determined via the following recursive formulae:

$$\begin{aligned} C_1(v) &= C_{1R}(v), \quad v = 0, 1, \dots, K-1, \\ C_2(K) &= C_{2R}(K), \end{aligned}$$

where

$$C_{kr}(\nu) = C_{k,r-1}(\nu) - (1 - B_{k,r})x_r C_{k,r-1}(\nu - 1) + x_r C_{kr}(\nu - 1),$$

$$B_{k,r} = \begin{cases} g_r, & k = 1, \\ g_r y_r, & k = 2, \end{cases}$$

for $k = 1, 2$, $r = 2, \dots, R$, $\nu = 1, 2, \dots, K - 2 + k$, with initial conditions

$$C_{k1}(\nu) = \begin{cases} 1, & \nu = 0, \\ B_{k,1}x_1^\nu, & \nu = 1, 2, \dots, N - 2 + k, \end{cases}$$

$$C_{kr}(0) = 1 \quad r = 2, \dots, R,$$

for $k = 1, 2$.

Similarly, the utilisation U_i can be expressed as

$$U_i = \frac{1}{Z} \left(\sum_{\nu=1}^{K-1} C_1^{(i)}(\nu) + C_2^{(i)}(K) \right), \quad i = 1, 2, \dots, R, \quad (4.5)$$

where

$$C_k^{(i)}(\nu) = (1 - B_{k,i})x_i C_k^{(i)}(\nu - 1) + B_{k,i}x_i C_k(\nu - 1), \quad \nu = 2, \dots, K - 2 + k,$$

$$k = 1, 2, \quad i = 1, 2, \dots, R, \text{ with initial conditions } C_k^{(i)}(1) = B_{k,i}x_i.$$

The marginal state probabilities $\{\hat{p}_i(\ell_i): \ell_i = 0, \dots, K\}$ can be determined by using ME solution (4.1) and the recursive expressions for $C_k^{(i)}(\nu)$. Let $n(i)$ be the random variable for the number of cells at queue i , $i = 1, 2, \dots, R$. Then the marginal state probabilities are given by (c.f. [14]).

$$p_i(\ell_i) = \Pr[n(i) \geq \ell_i] - \Pr[n(i) \geq \ell_i + 1], \quad (4.6)$$

where

$$\Pr[n(i) \geq \ell_i] = \frac{x_i^{\ell_i-1}}{Z} \left(\sum_{\nu=\ell_i}^{K-1} C_1^{(i)}(\nu - \ell_i + 1) + C_2^{(i)}(K - \ell_i + 1) \right),$$

$$i = 1, 2, \dots, R, \quad \ell_i = 1, 2, \dots, K.$$

Finally the aggregate state probabilities $\{p(n): n = 0, \dots, K\}$ are given by

$$p(n) = \begin{cases} \frac{1}{Z} & n = 0, \\ \frac{1}{Z} C_1(n), & n = 1, 2, \dots, K-1, \\ \frac{1}{Z} C_{21}(n), & n = K. \end{cases} \quad (4.7)$$

4.2 An ME Solution for the $S_{R \times R}(GE^R / GE / 1) / K$ Queueing System: An Extension

Earlier applications of entropy maximisation (e.g., [10, 12, 13, 17]) on arbitrary QNMs and shared buffer queues imply a decomposition into individual queueing systems with revised GE or GGeo-type interarrival and service time processes. These processes utilise analytic functions describing GE or GGeo-type flows amongst the queues of the network. Flows are split when going to different destinations and merged when converging from different sources. The formulae used to split flows are exact in the case of random routing. For GE or GGeo merging flows a two moment matching function is used to approximate the resultant stream with a GE or GGeo-type stream. This last operation may lead to some inaccuracies in extremal cases, where there are large differences in the size of the SCVs of the merging flows.

In this work, a ME QLD is proposed for an $S_{R \times R}(GE^R / GE / 1) / K$ queueing system which employs multiple input streams. This ME solution is of the same form as (4.1), subject to mean value constraints $\{U_i, \tilde{L}_i, \varphi_i; i = 1, 2, \dots, R\}$. The Lagrangian coefficients g_i and x_i of ME solution (4.1) are assumed to be invariant of the buffer size and are thus of the same form as these of a stable $GE^R / GE / 1$ queue (see Appendix I) i.e., $\{g_i, x_i; i = 1, 2, \dots, R\}$ are determined by making asymptotic connections with the ME solution of a stable $GE^R / GE / 1$ queue and, clearly, are given by

$$g_i = \frac{\rho_i(1-x_i)}{x_i(1-\rho_i)}, \quad x_i = \frac{\tilde{L}_i - \rho_i}{\tilde{L}_i}, \quad i = 1, 2, \dots, R, \quad (4.8)$$

where \tilde{L}_i is the MQL of a stable $GE^R / GE / 1$ queue (see Appendix I) and is given by

$$\tilde{L}_i = \frac{1}{2} \left(\rho_i + \frac{\sum_{j=1}^R \rho_{ji} C a_{ji}^2 + \rho_i^2 C s_i^2}{1 - \rho_i} \right), \quad (4.9)$$

with $C a_{ji}^2$ the SCV of stream i , $\rho_{ji} = \Lambda_{ji} / \mu_i$, $j = 1, 2, \dots, R$ and $\rho_i = \sum_{j=1}^R \rho_{ji}$.

Equating g_i and x_i of the ME solution of a stable $GE^R / GE / 1$ queue with those of a stable $GE/GE/1$ queue with overall (merged) interarrival parameters Λ_i and Ca_i^2 , the following relationship can be established:

$$\Lambda_i Ca_i^2 = \sum_{j=1}^R \Lambda_{ji} Ca_{ji}^2, \quad i = 1, 2, \dots, R. \quad (4.10)$$

Thus, the ME solution of a stable $GE^R / GE / 1$ queue can be considered as an ME solution of a stable $GE/GE/1$ with merged arrival processes having as parameters

$$\Lambda_i = \sum_{j=1}^R \Lambda_{ji}, \quad i = 1, 2, \dots, R, \quad (4.11)$$

and

$$Ca_i^2 = \sum_{j=1}^R \frac{\Lambda_{ji}}{\Lambda_i} Ca_{ji}^2, \quad i = 1, 2, \dots, R. \quad (4.12)$$

Note that expressions (4.11) and (4.12) turn out to be identical with those suggested in by Gelenbe and Pujolle [3]. Moreover, the interdeparture process of a stable $GE^R / GE / 1$ queue has a SCV given by (c.f. [10, 16])

$$Cd_i^2 = \rho_i(1 - \rho_i) + \rho_i^2 C_{si}^2 + (1 - \rho_i) Ca_i^2, \quad i = 1, 2, \dots, R. \quad (4.13)$$

Let $\{\pi_{ji}; i, j = 1, \dots, R\}$ be the CLPs of input streams $\{j\}$ at output ports $\{i\}$ of a shared buffer $S_{R \times R}(GE^R / GE / 1) / K$ queue. These probabilities can be obtained by using similar GE-type arguments as those applied in the case of the shared buffer $S_{R \times R}(GE / GE / 1) / K$ queue (c.f. [14]) and are given by

$$\pi_{ji} = \frac{1}{Z} (F_{ji}(K) + C_2(K)), \quad (4.14)$$

where (i, j) , $i, j = 1, 2, \dots, R$, is the j th flow to output port i , and

$$F_{ji}(K) = \delta_{ji} \sum_{v=0}^{K-1} C_1(v) (1 - \sigma_{ji})^{K-v} + (1 - \delta_{ji}) \sum_{v=1}^{K-1} C_1^{(i)}(v) (1 - \sigma_{ji})^{K-v}, \quad (4.15)$$

where

$$\delta_{ji} = \frac{r_{si}}{r_{si}(1 - \sigma_{ji}) + \sigma_{ji}},$$

with

$$r_{si} = \frac{2}{Cs_i^2 + 1}, \quad \sigma_{ji} = \frac{2}{Ca_{ji}^2 + 1}, \quad i, j = 1, 2, \dots, R, \quad K \geq 2.$$

Lagrangian coefficients $\{y_i; i = 1, 2, \dots, R\}$ of the $S_{R \times R}(GE^R / GE / 1) / K$ can be determined by using the flow balance conditions,

$$\sum_{j=1}^R \Lambda_{ji}(1 - \pi_{ji}) = U_i \mu_i, \quad i = 1, 2, \dots, R. \quad (4.16)$$

Substituting (4.14) into (4.16) the following system of R non-linear equations with R unknowns $\{y_i; i = 1, 2, \dots, R\}$, is obtained:

$$C_2^{(i)}(K) = \rho_i \sum_{v=1}^{K-1} C_1(v) - \sum_{j=1}^R \rho_{ij} F_{ij}(K) - \sum_{v=1}^{K-1} C_1^{(i)}(v), \quad (4.17)$$

for all $i = 1, 2, \dots, R$ and $K \geq 2$.

System (4.17) can be solved by applying the numerical algorithm of Newton-Raphson, which is generally expected to give quadratic convergence. One significant limitation of this method is the requirement that the partial derivatives of the Jacobian matrix must be calculated at each iteration. However, this requirement may be avoided by applying an efficient recursive scheme suggested in [14]. Thus, because of the recursive nature of the z -transforms which are used in the computational implementation of the ME solution, the $S_{R \times R}(GE^R / GE / 1 / K)$ queueing model can be used as an effective building block in the analysis of large MINs.

Note that in the special case of $R=2$, these equations (4.17) can be solved analytically yielding the following closed-form expressions

$$y_1 = \frac{1}{2g_1x_1} \left(\sqrt{A - B} \right), \quad (4.18)$$

where

$$\begin{aligned} A &= \left[\frac{x_1^K - x_2^K}{x_1^{K-1} - x_2^{K-1}} \right]^2 + \left[C_2^{(2)}(K) - C_2^{(1)}(K) \right]^2 \left[\frac{x_1 - x_2}{x_1^K - x_2^K} \right]^2 + 2 \left[C_2^{(2)}(K) + C_2^{(1)}(K) \right] \frac{x_1 - x_2}{x_1^{K-1} - x_2^{K-1}}, \\ B &= \frac{x_1^K - x_2^K}{x_1^{K-1} - x_2^{K-1}} - \left[C_2^{(2)}(K) - C_2^{(1)}(K) \right] \frac{x_1 - x_2}{x_1^K - x_2^K}, \quad \text{and} \\ y_2 &= \frac{y_1 g_1 x_1 (x_1^K - x_2^K) + \left(C_2^{(2)}(K) - C_2^{(1)}(K) \right) (x_1 - x_2)}{g_2 x_2 (x_1^K - x_2^K)}. \end{aligned} \quad (4.19)$$

Proofs of equations (4.18) and (4.19) are given in Appendix II.

5 ME ANALYSIS OF BANYAN MINs WITH ARBITRARY SWITCH SIZES

Consider a packet-switched finite buffered ATM switch with a Banyan MIN-based architecture depicted in Figure 3. The ATM switch consists of L levels and M stages and employs as basic building blocks R -input and R -output shared buffer switching elements ($R \times R$ crossbar switches).

Let switch- (l, m) denoting a switching element located at the l th level and m th stage of the MIN. Each output (input) port is connected to a output (input) pin. The input and output pins of each switching element are labelled (including the MIN's external input and output pins) as "input- k " and "output- k ", $k=0, 1, \dots, R-1$ from top to bottom, respectively. In regular Banyan MINs, where all switching elements are the same size, $M = \log_R N$, where R is the size of each switching element and N is the number of external inputs (or outputs). Regular Banyan MINs form an array of switching elements and in this case the number of switching elements in a row is referred to as the level L , where $L=N/R$.

The input/output ports of the MIN form an array of 'pins' which are indexed by a row then column. There are N pins at each stage. Each output pin is linked to a single down stream input pin at the next stage. Connections from output ports pins to input port pins can be made in an arbitrary way. These connections form the topology of the network and are represented in the forwards (FTM) and backwards (BTM) topology matrices. Note that in a Banyan MIN only one path exists between an external input pin and an external output pin. The FTM and BTM have both M columns and N rows representing the grid of output and input port pins, respectively. Element (n, m) holds the number of the input {output} port pin at the $(m+1)^{\text{th}}$ {(m-1)th} stage that is connected to output (input) port pin n at the m^{th} stage, respectively.

The traffic arriving at the external input pins of the MIN is assumed to be bursty and is represented by GE interarrival times. The service (transmission) times at the output ports are also assumed to be GE distributed with mean, $1/\mu_k$ and SCV, Cs_k^2 . The flow to external input pin k is parameterised by the overall mean arrival rate, $\hat{\Lambda}_k$ and the SVC of interarrival times, Ca_k^2 . Incoming cells traverse the network according to both the network's topology matrices and $\{r_{ks}\}_{N \times N}$, the routing probability matrix, where r_{ks} is the probability that a cell originating at external input pin k has external output pin s as its destination. Cells arrive in geometrically distributed bulks, with an average bulk size of $(Ca_k^2 + 1)/2$. Cells that arrive in the same bulk will take the same route across the MIN i.e. the routing decision is made on a per bulk basis. It is assumed that stage 0 switching elements at the input edges of the MIN may have infinite or finite capacity buffers, $\{K_{\ell 0}: \ell=0, 1, \dots, L-1\}$. Moreover, switching elements in the interior or last stage of the MIN each have a fixed finite capacity buffer, $\{K_{\ell m}: m=1, 2, \dots, M-1; \ell=0, 1, \dots, L-1\}$. A cell is lost if on arrival at a stage 0 switching element finds a full buffer. However, every cell that enters the MIN is guaranteed delivery to its destination. This constraint along with the finite buffers of internal switches, implies that the MIN internally operates a blocking mechanism, which in this paper is based on RS blocking (c.f., Introduction).

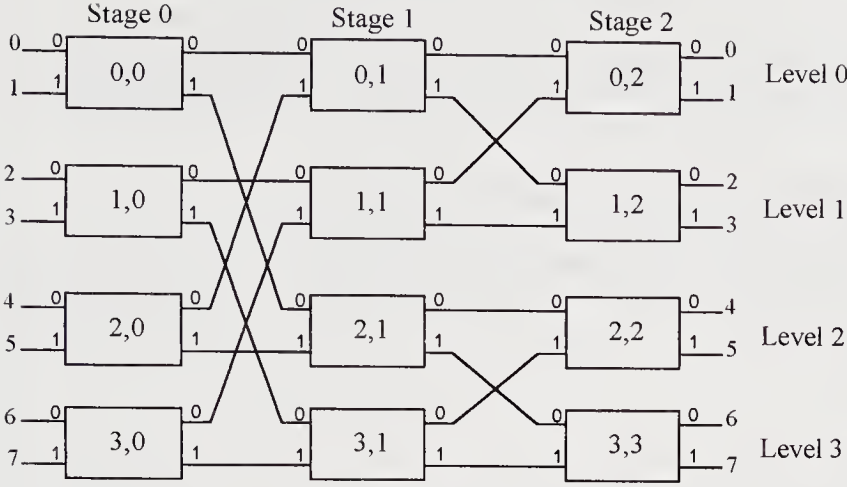


Figure 3 A 8x8 configuration of a regular Banyan Network

5.1 A ME Product Form Approximation

Suppose at any given time, the joint state of the network is denoted by $\mathbf{n}=(\mathbf{n}_{11},\dots,\mathbf{n}_{LM})$, where $\mathbf{n}_{ij}=(\mathbf{n}_{ij1},\mathbf{n}_{ij2},\dots,\mathbf{n}_{ijR})$ is the joint state of shared buffer queueing model of the switch- (i,j) and n_{ijk} is the number of cells queueing for output port k , $k=1,2,\dots,R$. Moreover, let $p(\mathbf{n})$ be at any given time the joint state probability of the network. The form of a ME solution, $p(\mathbf{n})$, of a Banyan MIN, subject to normalisation and the marginal constraints of shared buffer queueing systems used in Section 4, namely utilisation, U_{ijk} , $0<U_{ijk}<1$, ML , L_{ijk} , $U_{ijk}<L_{ijk}<K_{ij}$, and conditional aggregate full buffer probability with $n_{ijk}>0$, φ_{ijk} , $0<\varphi_{ijk}<1$, $j=1,2,\dots,R$, $i=1,2,\dots,LM$, is given - via the method of Lagrange's undetermined multipliers — as

$$p(\mathbf{n}) = \frac{1}{Z} \prod_{i=1}^L \prod_{j=1}^M \prod_{k=1}^R g_{ijk}^{s_{ijk}(\mathbf{n}_{ij})} x_{ijk}^{n_{ijk}} y_{ijk}^{f_{ijk}(\mathbf{n}_{ij})}, \quad (5.1)$$

where Z is the normalising constant and $\{g_{ijk}, x_{ijk}, y_{ijk}\}$, are the Lagrangian coefficients corresponding to constraints $\{U_{ijk}, L_{ijk}, \varphi_{ijk}\}$, respectively and $s_{ijk}(\mathbf{n}_{ij})$ and $f_{ijk}(\mathbf{n}_{ij})$ are appropriate indicator functions such that $s_{ijk}(\mathbf{n}_{ij})=1$, if $n_{ijk}>0$, or 0, otherwise and $f_{ijk}(\mathbf{n}_{ij})=1$, if $\sum_{k=1}^R n_{ijk} \leq K_{ij}$, or 0, otherwise, $k=1,2,\dots,R$. The form of ME solution (5.1)

clearly suggests a product form approximation, namely

$$p(\mathbf{n}) = \prod_{i=1}^L \prod_{j=1}^M p_{ij}(\mathbf{n}_{ij}), \quad (5.2)$$

where $p_{ij}(n_{ij})$ is determined by the ME solution (4.1) of each shared buffer queueing model.

The ME solution (5.1) can be implemented computationally by decomposing the network into individual building blocks of shared buffer switches-(i,j) with modified arrival and service parameters which capture the characteristics of the Banyan MIN.

5.2 Flow Through the Switching Elements of a Banyan MIN

The flow rate from each input pin through to each output pin of a Banyan network is calculated from the flow rate entering each input pin and the routing probability matrix $\{r_{ks}\}_{N \times N}$. Let $\hat{\lambda}_{ks}$ be the effective flow rate from external input pin k to external output pin s . Then, it follows that

$$\hat{\lambda}_{ks} = \hat{\Lambda}_k (1 - \pi_k) r_{ks} \quad k, s = 0, 1, \dots, N-1, \quad (5.3)$$

where $\pi_k = \pi_{ai}$, i is the input port of a switch at stage 0 that corresponds to input pin k and π_{ai} is the aggregate CLP of input port i , i.e., the probability that an arriving cell via external input k will be turned away (c.f. Section 6.1).

In Banyan networks only one path exists between k and s , so $\hat{\lambda}_{ks}$ is the contribution of flow given to each switch on the path from k to s . The effective flow rates, $\{\lambda_{ji}\}$, across input-output pin pairs $\{(j,i): i,j=1,2,\dots,R\}$ of a switching element can be obtained by appropriate summation of flows $\{\hat{\lambda}_{ks}\}$. For each input pin j , it is necessary to know the set of external input pins which connect to it (generally through other switches). Likewise, for each output pin i , it is necessary to know the set of external output pins which ultimately connect to it. Let these sets be denoted by $\text{Inpins}(j, m)$ and $\text{Outpins}(i, m)$, where (j, m) and (i, m) represent input pin j of a switching element and output pin i both at stage m , respectively. Any path that originates from an input pin in $\text{Inpins}(j, m)$ and terminates at a output pin in $\text{Outpins}(i, m)$ must pass through input j to output i . Thus, the effective flow rate from input pin j to output pin i , λ_{ji} , is given by

$$\lambda_{ji} = \sum_{\substack{k \in \text{Inpins}(j, m) \\ s \in \text{Outpins}(i, m)}} \hat{\lambda}_{ks}, \quad i, j = 1, \dots, R. \quad (5.4)$$

The method of calculating $\text{Inpins}(j, m)$ and $\text{Outpins}(i, m)$ is given in Appendix III.

Note that the shared buffer $S_{\text{RXR}}(\text{GE}^R / \text{GE} / 1) / K$ queueing model and product form approximation (5.1) are applicable to the performance analysis of packet switched finite buffered MINs with arbitrary configuration. However, in this more general case, there are more than one paths through the MIN, connecting an external input pin with an external output pin, and thus, some form of routing description is needed, in addition, to specify the flow.

6 ME ANALYSIS OF SWITCHING ELEMENTS WITHING THE BANYAN NETWORK

This section presents an approximate ME analysis of three types of shared buffer queueing models of switching elements within Banyan network, based on the $S_{R \times R}(GE^R / GE / 1) / K$ buliding block queue and GE-type flow formulae. Note that for presentational purposes, only subscripts for, input/output ports and related flow streams are denoted in this and subsequent section.

6.1 Case 1: Switching Elements at the Input Edges of the Network

When a switch is at the input edge of the Banyan network, the actual (overall) arrival parameters are known. However, due to potential RS blocking from second stage switching elements, the perceived (effective) service time (i.e., total transmission time experienced by each packet) has to be calculated. The effective service time can be expressed in terms of the blocking probabilities. A service completer which finds its downstream buffer full repeats its service. As each output port is connected to only one input pin of a downstream switching element, it is appropriate to calculate the effective service time in terms of the overall blocking probability that a service completer at output port queue i experiences at its downstream queue switch. This overall blocking probability is clearly given by

$$\pi_{ci} = \sum_{l=1}^R \pi_{kl} \frac{\Lambda_{kl}}{\tilde{\Lambda}_k}, \quad \tilde{\Lambda}_k = \sum_{l=1}^R \Lambda_{kl}, \quad i, k = 1, 2, \dots, R, \quad (6.1)$$

where k is the input pin of a switching element at the next stage which is connected with output pin i (defined in FTM), l is an output pin of the same element which is connected with k and Λ_{kl} is the overall arrival rate from input pin k to output pin l , $l = 1, 2, \dots, R$.

By considering GE-type probabilistic arguments, the effective service time parameters can be expressed by (c.f., [12])

$$\hat{\mu}_i = \mu_i (1 - \pi_{ci}) \quad i = 1, 2, \dots, R, \quad (6.2)$$

and

$$\hat{C}S_i^2 = \pi_{ci} + (1 - \pi_{ci})CS_i^2, \quad i = 1, 2, \dots, R. \quad (6.3)$$

The arrival rate from the external input stream j to output pin i , Λ_{ji} , is obtained by multiplying input rate, $\hat{\Lambda}_j$, by the sum of the appropriate routing probabilities (i.e. adding together the the probabilities from j to all external (destination) output pins that pass through i), namely

$$\Lambda_{ji} = \hat{\Lambda}_j \sum_{s \in \text{Outputs}(i, j)} r_{js}, \quad i, j = 1, 2, \dots, R. \quad (6.4)$$

Moreover, the SCV of the interarrival process from input stream j to input pin i is the same as that of the external SCV of interarrival time, as routing occurs on a per bulk basis (see Section 5) i.e.,

$$Ca_{ji}^2 = Ca_j^2, \quad i, j = 1, 2, \dots, R, \quad (6.5)$$

where Ca_j^2 is the SCV of the overall interarrival time at external input pin i .

For first stage switching elements with infinite capacity, the SCV of the interdeparture process is clearly given by (cf., (4.13) [10, 12])

$$Cd_i^2 = \hat{\rho}_i(1 - \hat{\rho}_i) + \hat{\rho}_i^2 \hat{Cs}_i^2 + (1 - \hat{\rho}_i)Ca_i^2, \quad i = 1, 2, \dots, R, \quad (6.6)$$

where

$$Ca_i^2 = \sum_{j=1}^R \frac{\Lambda_{ji}}{\Lambda_i} Ca_{ji}^2, \quad \text{and} \quad \hat{\rho}_i = \Lambda_i / \hat{\mu}_i, \quad i = 1, 2, \dots, R.$$

Note that in this case, each output port behaves as if it were an independent $GE^R / GE / 1$ queue with marginal ME QLD, $p_r(n_r), n_r = 1, 2, \dots, K$, given in Appendix I.

For first stage switching elements of finite capacity, the SCV of the interdeparture process is clearly given by (c.f., (4.13), [10, 12])

$$Cd_i^2 = \hat{\rho}_i(1 - \hat{\rho}_i) + \hat{\rho}_i^2 \hat{Cs}_i^2 + (1 - \hat{\rho}_i)\hat{Ca}_i^2, \quad i = 1, 2, \dots, R, \quad (6.7)$$

where

$$\begin{aligned} \lambda_{ji} &= \Lambda_{ji}(1 - \pi_{ji}), & \hat{Ca}_{ji}^2 &= \pi_{ji} + (1 - \pi_{ji})Ca_{ji}^2, & i, j &= 1, 2, \dots, R, \\ \lambda_i &= \sum_{j=1}^R \lambda_{ji}, & \hat{Ca}_i^2 &= \sum_{j=1}^R \frac{\lambda_{ji}}{\lambda_i} \hat{Ca}_{ji}^2 & i &= 1, 2, \dots, R, \end{aligned}$$

and

$$\hat{\rho}_i = \lambda_i / \hat{\mu}_i, \quad i = 1, 2, \dots, R,$$

The CLP π_{ji} can be determined from the ME solution of the shared buffer $S_{R \times R}(GE^R / GE / 1) / K$ queue (c.f., Section 4.2), namely,

$$\pi_{ji} = \frac{F_{ji}(K) + C_2(K)}{\sum_{v=0}^{K-1} C_1(v) + C_2(K)}, \quad (6.8)$$

where $F_{ji}(K)$ is given by equation (4.16) incorporating parameters $\Lambda_{ji}, Ca_{ji}^2, \hat{\mu}_i, \hat{C}s_i^2$, as appropriate.

The aggregate blocking probability, π_{aj} , at input pin j is clearly given by

$$\pi_{aj} = \sum_{i=1}^R \pi_{ji} \frac{\Lambda_{ji}}{\Lambda_j}, \quad j = 1, 2, \dots, R. \quad (6.9)$$

Note that $C_1(v)$ is a function of the Larangian coefficients $\{g_i, x_i; i = 1, 2, \dots, R\}$ (which can be calculated from the input parameters), whilst $C_2(K)$ is dependent upon all Lagrangian coefficients $\{g_i, x_i, y_i; i = 1, 2, \dots, R\}$. The $\{y_i\}$ coefficients are obtained by solving the non-linear equations which are of the same form as the ones determined by (4.17), if $R > 2$ or (4.18)-(4.19), if $R = 2$. The solution of these equations along with those of Section 4.2 give the QLDs of switching elements at stage 0 of the MIN together with other performance metrics.

6.2 Case 2: Switching Elements at the Interior of the Network

When a switching element is internal to the Banyan network at stage m , $m = 1, 2, \dots, M-1$, the throughput (effective arrival rate) can be determined in terms of the effective arrival rates of the external input ports, the routing probabilities and the network topology. The SCV of the effective interarrival process is obtained from the SCV of the output process of the previous stage. The values of the Lagrangian coefficients of the ME solution $p(\mathbf{n})$, $\mathbf{n} \in S(N, R)$, can be computed in terms of parameters of the overall flow which are related to the parameters of the effective flows and the blocking probabilities. These form a set of additional equations to those in Section 6.1 which (in addition) need to be solved to produce the QLD and other performance metrics for each internal switching element.

Let the effective flow rate that enters an input pin j be denoted by λ_j , $j = 1, 2, \dots, R$, with its component flow (j, i) going to output port i be denoted by λ_{ji} , $i = 1, 2, \dots, R$. Let $\hat{C}\alpha_{ji}^2$, $i, j = 1, 2, \dots, R$, be the SCV of flow (j, i) and π_{ji} , $i, j = 1, 2, \dots, R$ be the blocking probability that flow (j, i) will find a full buffer. Using these parameters, the overall flow from each input pin to each output port can be calculated, as follows:

The overall flow rate, Λ_{ji} , is clearly given by (c.f. [12])

$$\Lambda_{ji} = \frac{\lambda_{ji}}{1 - \pi_{ji}} \quad , \quad i, j = 1, 2, \dots, R, \quad (6.10)$$

and from the GE-type splitting flow formulae

$$Ca_{ji}^2 = \frac{\hat{C}a_{ji}^2 - \pi_{ji}}{1 - \pi_{ji}} \quad , \quad j, i = 1, 2, \dots, R, \quad (6.11)$$

where π_{ji} is calculated from the ME solution of the shared buffer $S_{R \times R}(GE^R / GE / 1) / K$ queue as described in Section (6.1).

The total effective arrival rate at input pin j , λ_j is expressed as

$$\lambda_j = \sum_{k=1}^R \lambda_{kj}, \quad j = 1, 2, \dots, R,$$

whilst the transition probability of a job going from input pin i to output pin j is clearly given by

$$\alpha_{ji} = \frac{\lambda_{ji}}{\lambda_j}, \quad i, j = 1, 2, \dots, R.$$

Packets that arriving in the same batch follow the same route through the network. This means that within the network, splitting of departing flows (from individual servers) may be complex, but fall within two schemes. In the first scheme individual packets choose their own downstream queue, upon service completion, according to a Bernoulli filter. In the second scheme the routing decision is made on a per bulk basis where the head of the bulk (i.e. the first packet in the bulk) chooses its downstream queue according to a Bernoulli filter and subsequent members of the bulk follow in its path. The second scheme produces bigger arriving bulks than the first scheme. To this end, the effective SCV of the arrival process is determined from the GE-type splitting flow formulae, namely

$$\hat{C}a_{ji}^2 = 1 + (Cd_{pred(j)}^2 - 1)\alpha_{ji} \quad , \quad i, j = 1, 2, \dots, R, \quad (6.12)$$

where $Cd_{pred(j)}^2$ is the SCV of the interdeparture process from the upstream port/switch connected at stage $m-1$ to input pin j whose location is given by vector BTM (i, m) , $i=1, 2, \dots, R$, $m=1, 2, \dots, M-1$.

If the protocol indicates that the entire departing bulk will be directed to the same destination input port, then no splitting takes place and

$$\hat{C}a_{ji}^2 = C_{pred(j)}^2, \quad j = 1, 2, \dots, R. \quad (6.13)$$

Finally, the interdeparture process from output port i is given by (4.13), namely

$$Cd_i^2 = \hat{\rho}_i(1 - \hat{\rho}_i) + \hat{\rho}_i^2 \hat{C}s_i + (1 - \hat{\rho}_i) \hat{C}a_i^2, \quad i = 1, 2, \dots, R, \quad (6.14)$$

where

$$\hat{C}a_i^2 = \sum_{j=1}^R \frac{\lambda_{ji}}{\lambda_i} \hat{C}a_{ji}^2 \text{ and } \hat{\rho}_i = \lambda_i / \hat{\mu}_i, \quad i = 1, 2, \dots, R.$$

As only the effective arrival parameters are known, the overall arrival parameters are given in terms of the blocking probabilities, which are themselves given by equation (4.14). These equations together form $R \times R$ non-linear simultaneous equations with $R \times R$ unknowns (i.e. the π_{ji} 's). Writing these equations as functions of the π_{ji} 's gives

$$f_{ji} = \pi_{ji} - \frac{F_{ji}(K) + C_2(K)}{\sum_{v=0}^{K-1} C_1(v) + C_2(K)}, \quad i, j = 1, 2, \dots, R. \quad (6.15)$$

Assuming that the value of $C_2(K)$ is known, the equations are solved using Newton-Raphson's method to give the value of π_{ji} 's. After the π_{ji} 's are calculated, the $\{y_i\}$ coefficients are obtained by solving the non-linear equations (4.17), if $R > 2$ or (4.18), if $R=2$. In the case of $R > 2$, a new value for $C_2(K)$ is calculated. This process is repeated until there is no change in the value of $C_2(K)$. The solution of these equations along with those of Section 4.2 give the QLDs of switching elements internal to the MIN together with other performance metrics.

6.3 Case 3: Switching at the Output Edges

When a switching element is at the external edge of the MIN, then its performance analysis follows from the ME solution of the shared buffer $S_{R \times R}(GE^R / GE / 1) / K$ queueing model of an internal switching element, except that the mean rate and SCV of the service time of each output port i are, respectively, the actual parameters (μ_i, Cs_i^2) , $i = 1, 2, \dots, R$.

7 AN ME APPROXIMATION PROCEDURE FOR THE PERFORMANCE ANALYSIS OF BANYAN MINS

In this section an approximate procedure for obtaining the ME QLDs and other performance metrics at each building block of a Banyan MIN based shared buffer ATM switch is described. The procedure for infinite and finite first stage building blocks differ only in that the later

includes the calculation of the first stage blocking probabilities and the flow rates through the network which are depended upon these probabilities. It is assumed that and the interdeparture processes to be of GE type. When these processes split into a number of streams distributed, to different output ports, it is assumed that the splitting is Bernoulli. These assumptions give rise to interarrival processes which are approximately the superposition of GE streams. Thus interarrival processes can be determined and their parameters evaluated.

7.1 An ME Algorithm the Analysis of Banyan Networks

Begin

Step 1. Initialise all cell loss probabilities. Set SCV of inter-arrival times to 1;

Step 2. Calculate effective flows across Banyan MIN and at each switching element (c.f. section 5.2);

Step 3. At the first stage represent each of its switching elements as a shared buffer building block queue $S_{R \times R} \left(GE^R(\Lambda_{ji}, Ca_{ji}^2) / GE(\hat{\mu}_i, \hat{Cs}_i^2) / 1 \right) / \infty$, $i, j = 1, 2, \dots, R$, in the case of infinite capacity, or as $S_{R \times R} \left(GE^R(\Lambda_{ji}, Ca_{ji}^2) / GE(\hat{\mu}_i, \hat{Cs}_i^2) / 1 \right) / K$, $i, j = 1, 2, \dots, R$ for the case of finite capacity, and calculate for each output pin i the SCV of the interdeparture process Cd_i^2 , $i=1, 2, \dots, R$, to be used in the next stage using equations (6.6) and (6.7), as appropriate;

Step 4. From left to right do until last but one stage:

represent each stage switching element as a shared buffer building block queue $S_{R \times R} \left(GE^R(\lambda_{ji}, \hat{Ca}_{ji}^2) / GE(\hat{\mu}_i, \hat{Cs}_i^2) / 1 \right) / K$, $i, j = 1, 2, \dots, R$, and calculate for each output pin i the SCV of the interdeparture process Cd_i^2 , $i = 1, 2, \dots, R$, for the next stage, using equation (6.14)

Step 5. Analyse the performance of each switching element by solving a shared buffer building block queue $S_{R \times R} \left(GE^R(\lambda_{ji}, \hat{Ca}_{ji}^2) / GE(\mu_i, Cs_i^2) / 1 \right) / K$, $i, j = 1, \dots, R$.

For first stage switching elements with infinite capacity repeat Steps 3-5 and for the corresponding case of finite capacity repeat Steps 2-5 until convergence of the calculated values of the SCV of the interdeparture times and the blocking probabilities of the first stages (as appropriate). Print out ME QLDs and typical performance metrics.

End.

Remarks

The main computation effort of the ME algorithm is at every iteration between steps 3 and 5. The non-linear system of equations, $\{y_i: i=1, 2, \dots, R\}$, for each switching element can be written as $Y=F(Y)$, where Y and F are column vectors of dimension Ω , where Ω is the cardinality of the set $\{y_i\}$. Similarly the non-linear equations $\{\pi_{ij}: i, j=1, 2, \dots, R\}$ can be written as $\Pi=G(\Pi)$, where Π and G are column vectors of dimension Ω' , where Ω' is the cardinality of the set $\{\pi_{ij}\}$. It can be verified that the computational cost of the algorithm is $O(ML(\Omega^3 + \Omega'^3))$, where M is the number of stages and L is the number of levels of the MIN, Ω^3 is the number of

manipulations for inverting the Jacobian of F with respect to Y and Ω^{-3} is the number of manipulations for inverting the Jacobian of G with respect to Π .

The existence and unicity for the solution of the system of non-linear equations is difficult to prove analytically due to the complexity of the expressions of the blocking probabilities $\{\pi_{ij}\}$ and the expression of Lagrangian coefficients $\{y_i\}$. Furthermore no strict mathematical justification can be given for the convergence of $\{Cd_i^2: i = 1, 2, \dots, R\}$; nevertheless, numerical instabilities or non-convergence have never been observed in many experiments that have been carried out. If, however, at some iteration it is observed for at least one queue j that $\hat{\rho}_j = (\lambda_j / \mu_j) \geq 1$, then there exists only one trivial solution with $\pi_{ij} = 1, i \in \{1, 2, \dots, R\}$, which is outside the domain at validity of the model.

When switching elements of infinite capacity are present at the first stage 0, necessary conditions for the stability of the entire network are not obvious due to the constraining influence on a output port's service rate by downstream blocking. In essence, the stability condition for a single output port is that the effective arrival rate be less than its effective service rate which can only be approximately determined. This subject merits further research.

In cases of hot spot routing, cells are directed towards one particular output with a high probability. As this probability approaches unity, the MIN becomes equivalent to an arbitrary network with blocking and has an inverted tree configuration.

8 NUMERICAL RESULTS

This section presents typical numerical results in Tables 1-12 focusing on 8×8 (c.f., Tables 1-11) and 27×27 (c.f., Table 12) Banyan MINs with 2×2 and 3×3 switching elements, respectively. The aims of the study is to (i) validate the relative accuracy of the ME approximation algorithm against simulation (SIM) (c.f., Tables 1-8) (ii) define experimental bounds (c.f., Table 12) and, (iii) perform a buffer capacity optimisation across the entire Banyan Network (c.f., Tables 9-11).

In all experiments, external input ports of the Banyan MIN at stage 0 receive traffic with identical parameters. In total, three different routing schemes are adopted, namely uniform routing (regular traffic) towards the external output pins at final stage 2 (c.f., Tables 1-3, 6-12), and moderately or substantially biased routing towards an external output pin referred to as a warm spot (c.f., Table 4) or hot spot (c.f., Table 5), respectively. Note that in the case of uniform routing all switching elements belong to a particular stage will have the same output statistics. However, in the general case of non-uniform routing, switching elements within a stage will have different performance metrics. For each input port at stage 0, and without loss of generality, identical routing probabilities biased towards the warm or hot spot are used in Tables 4 and 5, respectively. As a consequence, switching elements at each stage of the decode tree (i.e., the tree composed from the routes connecting external input pins with the warm or hot spot external output port) will have identical performance metrics. Thus, in both cases of uniform and non-uniform routing, performance metrics are only reported once in Tables 1-12 respectively.

Tables 1-8 present a validation study of the ME algorithm against simulation which includes aggregate MQLs $\{L_i: i = 0, 1, 2\}$ at stages 0, 1 and 2, throughputs $\{\lambda_2\}$ of either a typical external output port under uniform routing (c.f., Tables 1-3, 6-8) and warm/hot spot external output port (c.f., Tables 4 and 5), and also the aggregate CLP of a typical switching element at stage 0. Moreover, Tables 6-8 display aggregate and marginal state probabilities for a typical 8x8 Banyan network under uniform routing. Note that the simulation results in Tables 1-8 were produced at 95% confidence intervals by using the Queueing Network Analysis Package (QNAP-2). It can be observed that the ME solutions are consistently comparable with those of simulation (SIM) for a wide range of parameterisation, including deterministic transmission times applicable to ATM switching elements. Note that confidence intervals are of small magnitude e.g., typically ± 0.01 for MQLs. Moreover, percentage differences for MQLs are generally less than 10% and error tolerances for state and blocking probabilities, (i.e., absolute differences between ME and SIM results) are less than 0.05. The accuracy of ME approximations begin to deteriorate as the value of SCVs increases. This can be attributed to further violation of renewability assumptions of the various flow in the network.

The ME algorithm is utilised in performing a buffer capacity assignment optimisation across the Banyan MIN (c.f. Tables 9-11). Given an overall buffer allocation for the entire network, it is possible to carry out buffer assignments to individual switching elements in order to optimize the throughput or the end-to-end delay. Three different buffer allocation policies are considered by assigning more of the allocated buffer capacity to the first, second and third stages, respectively. From Tables 9-11, it can be observed that by placing more of the buffer allocation at the first stage of the network, the throughput can be increased whilst the end-to-end delay is not adversely affected. This behaviour is intuitively correct since the CLP is smaller than in the other two cases, thus allowing more packets into the network.

Finally, Table 12 focus on 27x27 Banyan networks with 3x3 switching elements under regular traffic. Relative performance comparisons are carried out between the ME solutions produced incorporating the routing of entire bulks within the network (c.f., (6.14)) and SIM results produced using specially designed programs written in C. It can be seen that the analytic solutions for first stage MQL, $\{L_0\}$ and aggregate CLPs, $\{\pi_\alpha\}$, are comparable in accuracy to those of simulation, as in the examples of Tables 1-8 (n.b., both ME algorithm and simulations use identical external inputs at stage 0). However, the ME solutions define (experimentally) pessimistic bounds over the corresponding SIM results produced concerning the MQLs of output ports at stages 1 and 2. This behaviour is due to the fact that the ME approximation overestimates the size of the bulk transitions in the interior of the network, and, subsequently, the SCV of the interarrival time of each internal and last stage output port. The study of analytic performance bounds merits further research.

9 CONCLUSIONS

A cost-effective approximate algorithm, based on the principle of ME and the notion of system decomposition, is proposed for the performance analysis and prediction of packet-switched buffered Banyan MIN-based ATM switch architectures with arbitrary buffer and building block sizes, GE-type interarrival and service times and RS internal blocking. Analytic ME solutions

Table 1 Uniform Routing

Banyan MIN No.	Input Data $\{r_{ij}=0.125, i,j=0,1,\dots,7\};$ $N=8; \mu=1; K=9$			Output Statistics					
	Λ	Ca^2	Cs^2	L_0	L_1	L_2	λ_2	π_a	Method
1	0.5	3	3	3.1760	2.6261	2.3536	0.4387	0.1226	ME
				3.1100	2.7210	2.3530	0.4440	0.1211	SIM
2	0.5	5	5	3.4429	2.7746	2.3691	0.3869	0.2263	ME
				3.4620	2.9050	2.3300	0.3877	0.2229	SIM
3	0.5	7	7	3.5240	2.8164	2.3376	0.3502	0.2995	ME
				3.6990	2.9180	2.3020	0.3466	0.3020	SIM
4	0.5	11	11	3.5525	2.8112	2.2586	0.3020	0.3959	ME
				4.0030	2.8840	2.0940	0.2912	0.4175	SIM
5	0.5	15	15	3.5372	2.7735	2.1887	0.2711	0.4578	ME
				4.1590	2.8510	1.9650	0.2559	0.4862	SIM

for the QLD of a shared buffer $S_{RXR} (GE^R/GE/1)/K$ queue in conjunction with GE-type formulae for the first two moments of the effective service times and traffic flows in the network, play the rôle of effective building blocks in the decomposition process of the entire network. Numerical results are included to illustrate the relative accuracy of ME approximations against simulation, define experimental MQL bounds in the interior and last stage of the network and to investigate the buffer capacity optimisation across the entire MIN. This study has shown that the ME approximation algorithm is a credible analytic tool for the cost- effective performance modelling and optimisation of complex MINs represented by Banyan networks. The ME algorithm can be extended towards the approximate analysis of ATM switch architectures with space and service priorities. Moreover, closed form expressions for queueing models of ATM networks with both bursty and correlated traffic can be derived based on the stochastic analysis of single finite queues with batch renewal arrival processes (c.f.,[18]). Extensions of this kind are the subject of current study.

Table 2 Uniform Routing

Banyan No	Input Data $\{r_{ij}=0.125, i, j=0, 1, \dots, 7\};$ $N=8; \mu=1; K=9$			Output Statistics					
	Λ	Ca^2	Cs^2	L_0	L_1	L_2	L_2	λ_α	Method
10	0.3	7	7	2.1278	1.6379	1.3560	0.2451	0.1829	ME
				1.9580	1.5800	1.3510	0.2537	0.1497	SIM
11	0.5	7	7	3.5240	2.8164	2.3376	0.3502	0.2995	ME
				3.6990	2.9400	2.2480	0.2467	0.3020	SIM
12	0.7	7	7	4.7570	3.7642	2.9907	0.4081	0.4169	ME
				5.2130	3.7750	2.6850	0.3847	0.4502	SIM

Table 3 Uniform Routing

Banyan No.	Input Data $\{r_{ij}=0.125, i, j=0, 1, \dots, 7\};$ $N=8; \mu=1; K=5$			Output Statistics					
	Λ	Ca^2	Cs^2	L_0	L_1	L_2	λ_2	π_α	Method
13	0.5	5	0	1.600	1.111	0.8952	0.3253	0.3493	ME
				1.269	0.8553	0.8404	0.3000	0.2899	SIM
14	0.5	5	1	1.7507	1.2776	1.0493	0.3237	0.3526	ME
				1.4280	1.1590	1.0640	0.3463	0.3026	SIM
15	0.5	5	3	1.9570	1.5271	1.2600	0.3164	0.3671	ME
				1.9450	1.5920	1.2760	0.3254	0.3487	SIM
16	0.5	5	5	2.1163	1.7074	1.3837	0.3071	0.3859	ME
				2.3160	1.7810	1.2890	0.2965	0.4109	SIM

Table 4 Hot Spot Routing

Banyan No.	Input Data			Output Statistics					
	$\{r_{ij}=0.02, i=0,1,\dots,7, j=1,2,\dots,7\};$ $\{r_{ij}=0.86, i=0,1,\dots,7, j=0\};$ $N=8; \mu=1; K=9$								
	Λ	Ca^2	Cs^2	L_0	L_1	L_2	λ_2	π_α	Method
16	0.1	3	1	0.4705	0.9920	2.5333	0.6926	0.0078	ME
				0.4763	1.0240	2.6470	0.7062	0.0062	SIM
17	0.1	7	3	0.7625	1.6055	2.9622	0.6384	0.1076	ME
				0.7720	1.7230	3.0470	0.6431	0.1034	SIM

Table 5 Warm Spot Routing

Banyan No.	Input Data			Output Statistics					
	$\{r_{ij}=0.11, i=0,1,\dots,7, j=1,2,\dots,7\};$ $\{r_{ij}=0.23, i=0,1,\dots,7, j=0\};$ $N=8; \mu=1; K=9$								
	Λ	Ca^2	Cs^2	L_0	L_1	L_2	λ_2	π_α	Method
18	0.3	7	3	1.9505	1.6403	1.8753	0.4523	0.1805	ME
				1.5670	1.4740	1.8440	0.7180	0.1432	SIM
19	0.5	5	5	3.6797	4.2865	4.5339	0.6926	0.2472	ME
				4.0050	4.4510	4.0750	0.6663	0.2733	SIM

Table 6 First Stage QLDs

Input Data: $\{Ca^2=5, Cs^2=3, \Lambda=0.5, \mu=1, K=5, N=8\};$
 $\{r_{ij}=0.125, i, j=0,1,\dots,7\}$

n	Aggregate QLD		Marginal QLD	
	ME	SIM	ME	SIM
0	0.3770	0.3565	0.6351	0.5864
1	0.1230	0.1412	0.1025	0.1417
2	0.1190	0.1300	0.0849	0.1111
3	0.1130	0.1189	0.0694	0.0793
4	0.1070	0.1064	0.0558	0.0501
5	0.1610	0.1470	0.0525	0.0313

Table 7 Second Stage QLDs

Input Data: $\{Ca^2=5, Cs^2=3, A=0.5, \mu=1, K=5, N=8\};$
 $\{r_{ij}=0.125, i, j=0, 1, \dots, 7\}$

<i>n</i>	Aggregate QLD		Marginal QLD	
	ME	SIM	ME	SIM
0	0.4220	0.3904	0.6589	0.6211
1	0.1740	0.1859	0.1346	0.1605
2	0.1360	0.1420	0.0887	0.1006
3	0.1040	0.1082	0.0576	0.0609
4	0.0790	0.0804	0.0364	0.0348
5	0.0840	0.0931	0.0237	0.0222

Table 8 Third Stage QLDs

Input Data: $\{Ca^2=5, Cs^2=3, A=0.5, \mu=1, K=5, N=8\};$
 $\{r_{ij}=0.125, i, j=0, 1, \dots, 7\}$

<i>n</i>	Aggregate QLD		Marginal QLD	
	ME	SIM	ME	SIM
0	0.4570	0.4572	0.6814	0.6734
1	0.2000	0.2019	0.1479	0.1575
2	0.1370	0.1368	0.0841	0.0858
3	0.0920	0.0894	0.0471	0.0459
4	0.0600	0.0583	0.0257	0.0236
5	0.0530	0.0564	0.0138	0.0137

Table 9 Buffer Assignment Biased for Stage 0

Input Data: $\{Ca^2=Cs^2=5, A=0.1, \mu=1, N=8\};$
 $\{r_{ij}=0.125, i, j=0, 1, \dots, 7\}$

<i>K</i> ₀	<i>K</i> ₁	<i>K</i> ₂	End-to-End Delay	CLP
9	9	9	9.0565	0.1170
11	8	8	9.5461	0.0838
13	7	7	9.9613	0.0616
15	6	6	10.3868	0.0472
17	5	5	10.9864	0.0392
19	4	4	12.1656	0.0384
21	3	3	15.3603	0.0541

Table 10 Buffer Assignment Biased for Stage 1

Input Data: $\{Ca^2 = Cs^2 = 5, A=0.1, \mu=1, K=5, N=8\};$
 $\{r_{ij}=0.125, i,j=0,1,\dots,7\}$

K_0	K_1	K_2	End-to-End Delay	CLP
9	9	9	9.0565	0.1170
8	11	8	8.7827	0.1380
7	13	7	8.4496	0.1648
6	15	6	8.0488	0.1984
5	17	5	7.5768	0.2401
4	19	4	7.0342	0.2923
3	21	3	6.4185	0.3585

Table 11 Buffer Assignment Biased for Stage 2

Input Data: $\{Ca^2 = Cs^2 = 5, A=0.1, \mu=1, K=5, N=8\};$
 $\{r_{ij}=0.125, i,j=0,1,\dots,7\}$

K_0	K_1	K_2	End-to-End Delay	CLP
9	9	9	9.0565	0.1170
8	8	11	8.7324	0.1415
7	7	13	8.3474	0.1716
6	6	15	7.8941	0.2088
5	5	17	7.3679	0.2553
4	4	19	6.7560	0.3142
3	3	21	6.0386	0.3904

APPENDIX I DERIVATION OF AN ME QLD FOR A STABLE $GE^R/GE/1$ QUEUE

Consider a stable FCFS $GE^R/GE/1$ single server queue depicted in Figure 4. The queue receives a multiple input of R streams with GE-type interarrival parameters.

$(\Lambda_{ji}, Ca_{ji}^2), j=1,2,\dots,R$. Moreover, the server provides GE-type service time with parameters (μ_i, Cs_i^2) .

Table 12 Analytical Bounds over Simulation for 27x27 MINs

Banyan No.	Input Data $\{r_{ij}=0.037, i,j=0,1,...,26\};$ $N=27;\mu=1; K=9$			Output Statistics				
	Λ	Ca^2	Cs^2	L_0	L_1	L_2	π_α	Method
18	0.10	1	1	0.3330	0.3330	0.3330	0.0000	ME
				0.3339	0.3331	0.3333	0	SIM
19	0.25	1	1	0.9999	0.9999	0.9999	0.0001	ME
				0.9969	1.0330	0.9997	0.0001	SIM
20	0.10	7	1	0.8925	0.9937	0.8475	0.1241	ME
				0.9000	0.5600	0.4800	0.1234	SIM
21	0.25	7	1	1.9750	2.2800	1.8069	0.1973	ME
				2.2020	1.3600	1.2203	0.2022	SIM
22	0.10	3	7	0.7200	0.7021	0.6927	0.0112	ME
				0.7541	0.6918	0.6378	0.0126	SIM
23	0.25	7	3	2.1780	2.4270	1.9477	0.1921	ME
				2.3981	1.8912	1.5398	0.2091	SIM

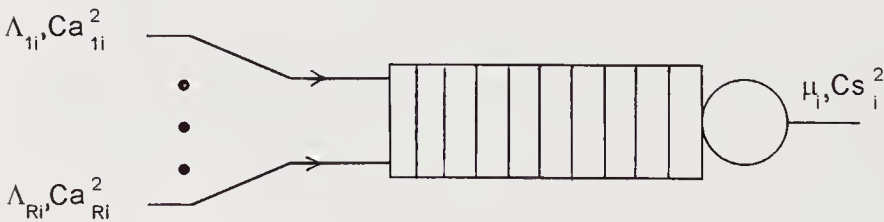


Figure 4. A stable $(GE^R / GE / 1)$ Queue i

Suppose all that is known about the $GE^R / GE / 1$ queue is that the server utilisation, ρ_i , and MQL, \tilde{L}_i . Entropy maximisation, subject to normalisation, utilisation and MQL constraints, implies that the QLD of the $GE^R / GE / 1$ queue is given by

$$p_i(n_i) = \frac{1}{Z_i} g_i^{h(n_i)} x_i^{n_i}, \quad (A1)$$

where $Z_i = 1 / p_i(0)$ is the normalising constant, $h(n_i)$ is an auxiliary function defined by $h(n_i) = 1$, if $n_i > 0$ or, 0, otherwise, and $\{g_i, x_i\}$ are the Langrangian coefficients corresponding to utilisation and MQL constraints, respectively.

The server utilisation, ρ_i is clearly expressed by

$$\rho_i = \sum_{j=1}^R \rho_{ji}, \quad \rho_{ji} = \Lambda_{ji} / \mu_i, \quad j = 1, 2, \dots, R. \quad (A2)$$

Moreover, an expression for the MQL, \tilde{L}_i , can be obtained from the generalised P-K expression for a stable $MB/G/1$ queue [9], namely

$$\tilde{L}_i = \frac{\rho_i}{2} + \frac{1}{2(1 - \rho_i)} \left(\rho_i^2 C s_i^2 + \rho_i b_i (C b_i^2 + 1) \right), \quad i = 1, 2, \dots, R, \quad (A3)$$

where b is the mean and $C b^2$ is the SCV of the bulk size distribution.

In the case of a number of arriving bulk Poisson streams with parameters b_{ji} and $C b_{ji}^2$, $j = 1, 2, \dots, R$, respectively, the overall arrival stream is another bulk Poisson stream with mean, b_i , and SCV, $C b_i^2$. The later parameters can be determined via the law of total moments, namely,

$$b_i = \sum_{j=1}^R b_{ji} p_{ji}, \quad i = 1, 2, \dots, R, \quad (A4)$$

where

$$p_{ji} = \frac{\Lambda_{ji}}{\Lambda_i}, \quad \Lambda_i = \sum_{j=1}^R \Lambda_{ji}, \quad i = 1, 2, \dots, R,$$

$$C b_i^2 = \frac{b_i^{(2)} - b_i^2}{b_i^2}, \quad i = 1, 2, \dots, R, \quad (A5)$$

where $b_i^{(2)}$ is the second moment of the bulk size of the overall stream, namely

$$b_i^{(2)} = \sum_{j=1}^R b_{ji}^{(2)} p_{ji}, \quad i = 1, 2, \dots, R, \quad (\text{A6})$$

where $b_{ji}^{(2)}$ is the second moment of the bulk size for stream j .

Manipulations lead to relations

$$(Cb_i^2 + 1)\Lambda_i b_i = \sum_{j=1}^R (Cb_{ji}^2 + 1)\Lambda_{ji} b_{ji}, \quad i = 1, 2, \dots, R. \quad (\text{A7})$$

Substituting into the generalised P-K expression (A3) the following formula for the MQL of a stable $M^B / G / 1$ queue with an aggregate of R multiple input streams is obtained:

$$\tilde{L}_i = \frac{\rho_i}{2} + \frac{1}{2(1 - \rho_i)} \left(\rho_i^2 C_{si}^2 + \sum_{j=1}^R \rho_{ji} b_{ji} (Cb_{ji}^2 + 1) \right), \quad i = 1, 2, \dots, R. \quad (\text{A8})$$

Note that the superposition of R GE-type streams results into an overall bulk Poisson process, but the bulk size distribution is determined by a sum of geometrics. Moreover, the individual parameters of each bulk size distribution b_{ji} and Cb_{ji}^2 can be expressed by

$$b_{ji} = \frac{Ca_{ji}^2 + 1}{2}, \quad i, j = 1, 2, \dots, R, \quad (\text{A9})$$

and

$$Cb_{ji}^2 = \frac{Ca_{ji}^2 - 1}{Ca_{ji}^2 + 1}, \quad i, j = 1, 2, \dots, R. \quad (\text{A10})$$

Using expressions (A9) and (A10), formula (A8) becomes identical for MQL expression (4.9). Moreover, substituting (A1) into the constraints of utilisation, ρ_i and MQL, \tilde{L}_i , and carrying out some manipulations, Lagrangian coefficients g_i and x_i are determined via expressions (4.14).

APPENDIX II DERIVATION OF THE LARGRANGIAN COEFFICIENTS $\{y_i: i = 1, 2, \dots, R\}$ FOR THE CASE OF $R=2$

The Largrangian coefficients $\{y_i: i = 1, 2, \dots, R\}$ of the $S_{R \times R}(GE^R / GE / I) / K$ are determined by solving the set of nonlinear simutaneous equation (4.17). These can be written as

$$C_2^{(i)}(K)[Y] = C_2^{(i)}(K), \quad i = 1, 2, \dots, R, \quad (A11)$$

where $[Y]$ denotes the vector of y_i 's. In the case of $R = 2$, equation (A11) can be solved analytically as follows:

$$\begin{aligned} C_2^{(1)}(K)[Y] = & g_1 x_1 y_1 (x_1^{K-1} + x_1^{K-2} x_2 + \dots + x_2^{K-1}) \\ & + g_2 x_2 y_2 g_1 x_1 y_1 (x_1^{K-2} + x_1^{K-3} x_2 + \dots + x_2^{K-2}), \end{aligned} \quad (A12)$$

and

$$\begin{aligned} C_2^{(2)}(K)[Y] = & g_2 x_2 y_2 (x_1^{K-1} + x_1^{K-2} x_2 + \dots + x_2^{K-1}) \\ & + g_2 x_2 y_2 g_1 x_1 y_1 (x_1^{K-2} + x_1^{K-3} x_2 + \dots + x_2^{K-2}), \end{aligned} \quad (A13)$$

which leads to

$$\begin{aligned} C_2^{(2)}(K)[Y] - C_2^{(1)}(K)[Y] = & g_2 x_2 y_2 (x_1^{K-1} + x_1^{K-2} x_2 + \dots + x_2^{K-1}) \\ & - g_1 x_1 y_1 (x_1^{K-1} + x_1^{K-2} x_2 + \dots + x_2^{K-1}). \end{aligned} \quad (A14)$$

Equation (A.14) can be simplified by using the identitiy

$$(x_1^{K-1} + x_1^{K-2} x_2 + \dots + x_2^{K-1})(x_1 - x_2) = (x_1^K - x_2^K).$$

To this end, solving (A.14) with respect to the Largrangian coefficient y_2 , it follows that (4.19) holds.

Substituting y_2 into (A.14), the following equation is obtained:

$$\begin{aligned} 0 = & y_1^2 g_1^2 x_1^2 (x_1^{K-1} - x_2^{K-1}) \\ & + y_1 \left(\frac{g_1 x_1 (x_1^K - x_2^K)^2 + g_1 x_1 (x_1^{K-1} - x_2^{K-1}) (C_2^{(2)}(K) - C_2^{(1)}(K))}{(x_1^K - x_2^K)} \right) - C_2^{(1)}(K). \end{aligned} \quad (A15)$$

Solving equation (A16) for y_1 and taking the positive root yields expression (4.18).

APPENDIX III METHOD OF CALCULATING THE INPINS AND OUTPINS SETS

Let $\text{Inpins}[i,m]$ be the set of external input pins of the Banyan Network which are connected to an interior input-port pin at position (i,m) of the array of input pins. Likewise let $\text{Inpins}'[i,m]$ be the set of external input pins that are linked to an interior output pin at position (i,m) of the array of output pins. Let S be the set of input pins that constitute the inputs of a particular switch. Each switch fully connects all its input-pins to its output pins, therefore the sets $\text{Inpins}[i,m]$ and $\text{Inpins}'[i,m]$ are related as follows:

$$\text{Inpins}'[i,m] = \bigcup_{k \in S} \text{Inpins}[k,m].$$

Each interior input-port pin is connected to one output pin from the previous stage, which can be determined from the backwards topology matrix (BTM), i.e., input-pin at position (i,m) connects with output pin located at $\text{BTM}[i,m]$, which is of course at stage $(m-1)$.

Thus, each input pin 'inherits' its Inpin set from the output pin that it is connected to, i.e., $\text{Inpins}[i,m] = \text{Inpins}'[\text{BTM}[i,m], m-1]$.

The input pins at the input edge of the network have only one element in their Inpins set, as they correspond to a particular external input pin, i , $i = 0, 1, \dots, N-1$ i.e.,

$$\text{Inpins}[i,0] = \{i\}.$$

Thus Inpins sets at each level are obtained via the following procedure:

```

for i = 0 to N-1
     $\text{Inpins}[0,i] = \{i\}.$ 
for M = 1 to M-1
    for i = 0 to N-1
         $\text{Inpins}'[i,m-1] = \bigcup_{k \in S} \text{Inpins}[k,m-1]$ 
    end i
    for i = 0 to N-1
         $\text{Inpins}[i,m] = \text{Inpins}'(\text{BTM}[i,m], m-1)$ 
    end i
end j

```

The method of calculating the Outpins sets is similar to the one for determining the Inpins sets but is applied in reverse order.

REFERENCES

- [1] Boyer P, Lehnert M R and Kuehn P J, "Queueing in an ATM basic switch element", Technical Report CNET-123-030-CD-CC, CNET, France, 1988.
- [2] Eng K Y, Karol M K and Yeh Y-S, "A Growable Packet (ATM) Switch Architecture: Design Principles and Applications," *IEEE Trans. Comm.*, **40**, no. 2, pp. 423-430, Feb. 1992.
- [3] Gelenbe, E. and Pujolle, G., The Behaviour of a single Queue in a General Queueing Network, *Acta Informatica*, **7**, pp. 123-160, 1976.
- [4] Harrison P G and Pinto A de C, "Blocking in Asynchronous, Buffered Banyan Networks," In Proc. of the *IFIP WG 7.3 International Conference on the Performance of Distributed Systems and Integrated Communication Networks*, Kyoto, Japan, ed. T Hasegawa, H Takagi and Y Takahashi, North-Holland, pp. 169-188, 1992.
- [5] Harrison P G and Pinto A de C, "An Approximate Analysis of Asynchronous, Packet-switched Buffered Banyan Networks with Blocking," *Performance Evaluation*, **19**, pp. 223-258, 1994.
- [6] Hong S, Perros H G and Yamashita H, "A discrete-time queueing model of the shared buffer switch with bursty arrivals," Research Report, Computer Science Department, North Carolina State University, 1992.
- [7] Jaynes, Prior probabilities, *IEEE Trans. Syst. Sci. Cybern.* SSC-4, pp.227-241, 1968.
- [8] Karol M J and C-L I, "Performance Analysis of a Growable Architecture for Broad-Band Packet (ATM) Switch," *IEEE Trans. Comm.*, **40**, no. 2, pp. 431-439, Feb. 1992.
- [9] Kleinrock, Queueing Systems, Vol.1: Theory, John Wiley and Sons, Inc., 1975.
- [10] Kouvatsos, A Universal Maximum Entropy Algorithm for the Analysis of General Closed Networks, *Computing Networking and Performance Evaluation*, IFIP WG 7.3, T. Hasegawa, et al (Eds.), pp.113-124, North Holland, 1986.
- [11] Kouvatsos, A Maximum Entropy Analysis of and the G/G/1 Queue at Equilibrium, *J. Opl. Res. Soc.*, Vol.39, pp.183-200, 1988.
- [12] Kouvatsos D and Xenios N, "MEM for Arbitrary Queueing Networks with Multiple General Servers and Repetitive-service Blocking," *Performance Evaluation*, **10**, pp. 169-195. Sep. 1989.
- [13] Kouvatsos and N. Tabet-Aouel, Product-Form Approximations for an Extended Class of General Closed Queueing Network, *Performance '90*, IFIP WG 7.3 and BCS, P. King et al (Eds.), North-Holland, pp.301-315, 1990.

- [14] Kouvatso D and Denazis S G, "A Universal Building Block for the Approximate Analysis of a Shared Buffer ATM Switch Architecture," *Annals of OR*, **44**, pp. 241-278, 1994.
- [15] Kouvatso D.D., Tabet-Aouel, N. and Denazis, S.G., "ME-Based Approximations for General Discrete-Time Queueing Models", Performance Evaluation, Special Issue on Discrete-Time Models and Analysis Methods of Performance Evaluation, Vol.21, pp81-109, 1994.
- [16] Kouvatso D, Entropy Maximisation and Queueing Network Models, *Annals of Operations Research*, Special Issue on Queueing Networks, Vol. 48, pp.63-126, 1994.
- [17] Kouvatso D.D. and Wilkinson, J., "A Product-Form Approximation for Discrete-Time Arbitrary Networks of ATM Switch Architectures", Performance Modelling and Evaluation of ATM Networks, IFIP Publication, Chapman and Hall, London, Vol.1, pp. 365-383, 1995.
- [18] Kouvatso D.D. and Fretwell, R. "Closed Form Performance Distributions of a Discrete Time $GI^G/D/1/N$ Queue with Correlated Traffic", Data Communications and their Performance, IFIP Publication, Fdida, S. and Onvural, R.O. (eds.), Chapman and Hall, London pp. 142-163, 1995.
- [19] Kuwahara H, Endo N, Ogino M and Kozaki T, "A Shared Buffer Memory Switch for an ATM Exchange," In *Proc. Int. Conf. on Communications*, Boston, MA, pp. 441-444, June 1989.
- [20] Lin, T., and Kleinrock L, "Performance Analysis of Finite-Buffered Multistage Interconnection Networks with a General Traffic Pattern," In *Proc. ACM SIG-METRICS '91*, pp. 68-89, 1991.
- [21] Nojo and Watanabe H, A New Stage Method Getting Arbitrary Coefficient of Variation by two stages, *Trans. IEICE* 70, pp. 33-36, 1987.
- [22] Shore, J.E., and Johnson, R.W., Axiomatic derivation of the principle of ME and the principle of minimum cross-entropy, *IEEE Trans. Info. Theory* IT-26, 1980.
- [23] Sauer, Configuration of Computing Systems: An Approach Using Queueing Network Models, PhD Thesis, University of Texas, 1975.
- [24] Szymanski T and Shaikh S, "Markov Chain Analysis of Packet-Switched Banyans with arbitrary Switch Sizes, Queue Sizes, Link Multiplicities and Speedups," *1989 IEEE INFOCOM*, pp. 960-971, 1989.
- [25] Tobagi F, "Fast Packet Switch Architectures for Broadband Integrated Services Digital Networks," *Proc. of the IEEE*, vol. 78, no. 1, Jan. 1990.

- [26] Yamashita H, Perros H G and Hong S, "Performance modelling of shared buffer ATM switch architecture," In *Teletraffic and Datatraffic in a Period of Change*, eds. Jensen and Iversen, North-Holland, 1991.

BIOGRAPHIES

Demetres Kouvatsos received the BSc degree in Mathematics from Athens National University in 1970, the MSc degree in Statistics from Victoria University of Manchester in 1971 and the PhD degree in Computation from UMIST, Institute of Science and Technology, University of Manchester in 1974. He is currently a Reader in Computer Systems Modelling, Department of Computing, University of Bradford. Since early 80's, he pioneered new and cost-effective methodologies for the approximate analysis of arbitrary queueing network models of computer and communication systems. He has held a series of EPSRC (UK) and industrial research grants and is the author or co-author of many papers in the areas of queueing theory and systems performance modelling. He acted as the Chairman of the first four IFIP Workshops on "Performance Modelling and Evaluation of ATM Networks" (1993-96), and the Co-Chairman of the 3rd International Workshop in Queueing Networks with Finite Capacity" (1995).

Jeffrey Wilkinson received the BSc degree in Computer Science from Bradford University in 1992 and, subsequently joined the Computer Systems Performance Modelling Group, Bradford University, as PhD research student. Since January 1996 he is employed as a Research Assistant at the Department of Computing, University of Bradford, under an EPSRC (UK) funded project on the performance modelling of ATM networks. His research interests include queueing theory and performance modelling of computer and communication systems.

Peter Harrison is currently a Reader in Computing Science at Imperial College where he became a Lecturer in 1983. He graduated at Christ's College Cambridge as a Wrangler in Mathematics in 1972 and went on to gain Distinction in Part III of the Mathematical Tripos in 1973, winning the Mayhew prize. He obtained his PhD in Computing Science at Imperial College in 1979. He has researched into analytical performance modelling techniques and algebraic program transformation for some fifteen years, visiting IBM Research Centers for two summers in the last decade. He has written two books, had over 70 research papers published in the areas of performance modelling and functional programming and held a series of research grants.

Madhu Bhabuta received her MEng degree from Imperial College in 1994. She was awarded a scholarship from the ESPRIT Basic Research group in November 1994 to fund work leading to a PhD degree. She is currently a Research Assistant at Imperial College and is supported by EPSRC. Her areas of interest include performance modelling, ATM traffic modelling and stochastic process algebras.

PART SEVEN

Bandwidth and Admission Control

Markov Chain Animation Technique Applied to ATM Bandwidth Derivation and Tandem Switches

J.M.Griffiths and J.M.Pitts

Queen Mary and Westfield College, University of London

Mile End Road, London, E1 4NS, United Kingdom

Tel. +44 171 975 5596, Fax. +44 181 981 0259

email J.M.Griffiths@qmw.ac.uk

Abstract

The derivation of queue lengths and buffer fills in ATM networks with particular types of cell arrival is known explicitly for some specific cases, but often simulation must be resorted to, which is time consuming and the accuracy is difficult to determine, particularly when one is considering cell losses or CDV around the 10^{-8} level. We describe an alternative technique which involves iteration of the Markov transition and state probabilities for a queue; we have called this Animation. Its principle is simply that of repetitively applying the appropriate Markov chain transition matrix to the current queue length probabilities to produce a new set of probabilities. In the case of a known cell sequence arrival the transition matrix changes for each iteration to reflect the arrival probability. Animation presents some numerical difficulties and techniques are described to overcome these. The specific application described is the derivation of equivalent bandwidth of cell patterns resulting from ATM policing or shaping, and the transit of these cells through networks containing multiple switching stages.

Keywords

ATM, bandwidth, policing, queuing, switching

1. INTRODUCTION

In connection with studies in policing functions for ATM [1] we had to consider their limitations and, in particular, the most adverse patterns (MAP) of full cells which would meet the policing criteria and hence could be applied to the network by users. To give an idea of the problem a typical MAP would be 10 full, 25 empty, 11 full, 26 empty, 11 full, 26 empty, 5 full and 133 empty cells. After this the sender could repeat the pattern. For different system parameters the number of full cells in each burst

could be much reduced and the number of bursts could be many more; the number of final empty cells may rise to several thousand. In order to assess the impact of a stream with such a MAP we needed to estimate their equivalent bandwidth and the method chosen was to add the stream to a infinite number of random sources with a Poisson load of 0.5. The resulting buffer fill probability would then be compared with a pure Poisson load adjusted to give a similar distribution in the area of probabilities of 10^{-8} . The excess of this second Poisson load over the background Poisson load of 0.5 would be regarded as the equivalent bandwidth of the MAP. To avoid the introduction of an arbitrary parameter the buffer was assumed to be infinite. A Poisson background load seemed appropriate in the absence of any clear understanding of the actual statistics; in the circumstances where there is traffic from a large number of sources ranging from CBR to bursty, central limit considerations suggested this choice.

Although the concept appeared reasonable its evaluation was not obvious. Analytic solutions to the general Poisson + pattern queuing problem are not known; the other alternative was simulation, but to get comparisons at the 10^{-8} level an impractical number of cells would be needed. From this need the technique of Markov Chain "animation" developed.

2. BACKGROUND

The Markov chain is well established as a conceptual device for describing the manner in which state probabilities are modified by events and their probabilities. In ATM terms the classic such chain is the buffer fed by a multiplicity of sources.

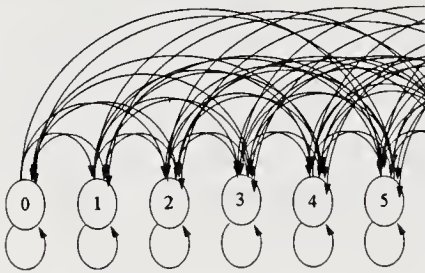


Figure 1 Markov Chain

In figure 1 the circles represent the states - in the case of the buffer they are the fills of the buffer, each with its associated probability s_0, s_1, s_2 , etc. The circular arrows represent the probability that after an event the machine remains in that same state s_{00}, s_{11}, s_{22} , etc. The curved arrows represent the probabilities of transition from one state to another. In the case of a buffer with a single server s_{11}, s_{22} , etc. will equal the probability, p_1 , that one cell arrives.

In the case of s_{00} this equals the probability that one or zero cells arrive. Similarly the probability that the machine moves from state n (n cells in buffer) to state $n + 1$ ($n + 1$ cells in buffer), i.e. s_{01}, s_{12}, s_{23} , etc. is the probability, p_2 , that 2 cells arrive; the probability that the machine moves from state n to state $n - 1$, i.e. s_{10}, s_{21}, s_{32} , etc. is the probability, p_0 , that no cells arrive.

The customary use for this information is to set up a set of equations to get an explicit solution to the steady state values of the state probabilities. This paper explores the animation method in which the Markov chain is iteratively applied to a set of data to derive solutions to queuing problems.

3. THE ANIMATION TECHNIQUE

In a machine with a finite number of states the initial state may be represented by a vector S of the form $[s_0, s_1, s_2, \dots]$. Multiplication by the transition matrix will produce the new vector S' representing the state probabilities after the first event (e.g. a batch of cells arriving and one being served). Thus one could start with an initial condition where no cells have arrived for a very long time, $s_0 = 1$ and $s_1, s_2, \dots = 0$. After the first event $s_0 = s_{00}$,

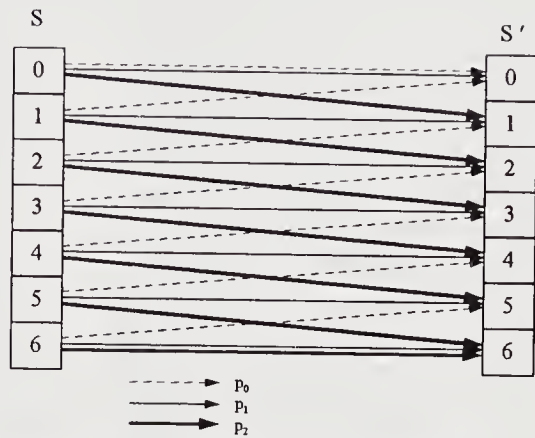


Figure 2 Simple Animation

$s_1 = s_{01}$, $s_2 = s_{02}$, etc. The next multiplication will establish the state after 2 events and so on. Continuing multiplication will lead S to tend to the steady state condition as calculated in the usual explicit way.

As an example of the process in action let us consider a queue of maximum length 6, being fed by two sources each with probability of full cell = 0.25, giving a total load of 0.5. This simplest of Binomial distributions gives the probability of an event with no full cells arriving, $p_0 = 0.5625$; with 1 cell, $p_1 = 0.375$; with 2 cells, $p_2 = 0.0625$. With a single cell served per event then $s_{11} = s_{22} = s_{33} = s_{44} = s_{55} = 0.375$. $s_{01} = s_{12} = s_{23} = s_{34} = s_{45} = s_{56} = 0.0625$. $s_{65} = s_{54} = s_{43} = s_{32} = s_{21} = s_{10} = 0.5625$. $s_{00} = 0.9375$ (the probability of no cells plus the probability of one cell). $s_{66} = 0.4375$ (the probability of one cell plus the probability of two cells). All other values in the transition matrix are zero as the corresponding transition is not possible. These probabilities can be repetitively applied to a starting state as shown in Figure 2. After 30 iterations stability is achieved to 6 decimal places accuracy. Effectively this process gives the transient state probabilities which are only rarely of interest. However it may be used as a means of finding the steady state probabilities in cases where an explicit solution is obscure. It has further use where the load (and hence the transition matrix) is fluctuating in a specific way, and this will be the subject of the core of this paper.

4. THE INFINITE QUEUE

In the introduction it was mentioned that derivation of the equivalent bandwidth was to be done by considering the probability distributions of an infinite queue loaded with:

- a background Poisson source + pattern
- a second purely Poisson source

An infinite queue was chosen to avoid introducing another arbitrary parameter but presents obvious computational difficulties as only finite queues can be handled numerically.

Extending figure 2 to the infinite case (figure 3) shows the difficulty; however large the actual queue length chosen, Z , with only the background source present one always needs one further element, Z^+ , to compute the next value of Z . Assuming that Z^+ is zero produces considerable distortion of the values in the upper range of the queue resulting in the need to compute a much extended queue length, with run time penalties. However a simple alternative results from the well known observation that the state probabilities closely approximate to a geometric series for low probabilities. Setting $Z^+ = Z^2/Y$ means that computation becomes apparently error free for practical purposes. This is indicated in figure 3 by the curved arrows.

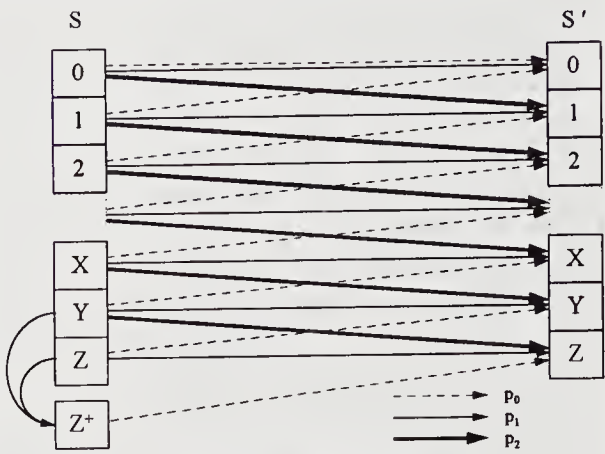


Figure 3 Infinite Queue

5. ROUNDING ERRORS

For a Poisson source of rate 0.5, calculation of the buffer fill probabilities by the explicit method or by continuous iteration results in obvious errors at low probabilities owing to rounding errors, even using double length arithmetic as shown in Table 1. It is clear that the figures are tending to a small (in this case negative) residual constant value rather than the expected value of zero. For the work to be described later it is necessary to continue the iterations considerably more than this initial simple example required. This leads to a residual constant of ever increasing magnitude which, if left uncorrected, would limit the utility of the technique. A solution to this problem also exploits the fact that the values should approximate a geometric series. Simple algebra can determine the value of the residual constant. If s_X, s_Y, s_Z are the values of the last three probabilities calculated and they are in a geometric progression with a constant error, ϵ , then

$$s_Z = s + \epsilon, \quad s_Y = \kappa s + \epsilon, \quad s_X = \kappa^2 s + \epsilon$$

Eliminating s and κ , the constant error is given by:

$$\epsilon = (s_X \cdot s_Z - s_Y^2) / (s_Z - 2 \cdot s_Y + s_X)$$

Queue Position	Uncorrected Probability	Corrected Probability
20	2.03E-11	
21	5.77E-12	
22	1.64E-12	
23	4.68E-13	
24	1.33E-13	
25	3.79E-14	
26	1.08E-14	
27	3.07E-15	
28	8.73E-16	8.75E-16
29	2.47E-16	2.49E-16
30	6.88E-17	7.09E-17
31	1.81E-17	2.02E-17
32	3.67E-18	5.74E-18
33	-4.35E-19	1.64E-18
34	-1.60E-18	4.65E-19
35	-1.94E-18	1.33E-19
36	-2.03E-18	3.77E-20
37	-2.06E-18	1.07E-20
38	-2.07E-18	3.06E-21
39	-2.07E-18	8.70E-22
40	-2.07E-18	2.48E-22

Table 1 Effect of Rounding Errors

The 3rd column in table 1 shows the effect if this is subtracted. The process works well until the denominator in the above expression becomes too small (values of s around 10^{-26}). In the case demonstrated the IEEE 64-bit floating point format was used with sign (1 bit), exponent (11 bits) and mantissa (52 bits). It is convenient to incorporate this correction into a normalisation procedure which is applied after every iteration; the procedure can also sum the queue probabilities to ensure that rounding errors do not cause the overall total to deviate from unity, neglecting the very small probability that the queue will exceed the length computed. Using this technique reduces computation time when low probability events are of interest as it avoids the use of extended length arithmetic.

6. ADDING STREAM CONSISTING OF AN ARBITRARY PATTERN

Figure 3 showed the transitions when the background load was present. If an extra full cell is present in the additional stream then the transitions are modified to those shown in figure 4. This is obviously not a stable situation as queue length probabilities will increase without limit. Interspersed empty cells are required in the additional stream to allow the queue length to reduce. Note that for simplicity of explanation we are still working with the case of 2 random sources to form the background load; the Poisson case works in exactly the same manner but of course the probabilities that more than two background cells arrive are

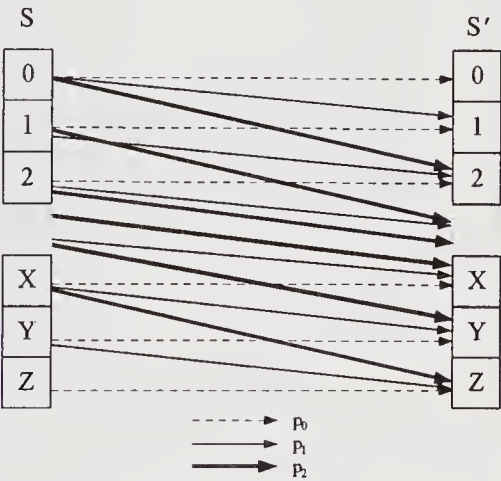


Figure 4 Background+Extra Cell

non - zero and so in the case of no extra full cell arriving $s'_n = \sum_{i=0}^{n+1} s_{n+1-i} p_i$ and if

there is an additional full cell $s'_n = \sum_{i=0}^n s_{n-i} p_i$ assuming that an arriving cell to an

empty queue can be served in the same time slot.

We now have all the components necessary to animate the buffer state probabilities. Starting from any initial set of state probabilities the appropriate transitions will be applied - in the rather simplified case we have been considering they will be as shown in figure 4 for every occurrence of a full cell in the additional stream, and figure 3 when there is no extra full cell in the additional stream. We will then end up with a new set of state probabilities. This can then be repeated using those final set of state probabilities as the new starting point. Eventually the final state probability vector will be the same as the starting vector. How many iterations this takes depends on how near the final answer is to the starting condition. An obvious choice is to start with the fill probabilities due to the background stream alone either derived iteratively as shown in section 3 above, or by an explicit solution.

Taking the simple example considered previously together with an additional stream of 1 full cell, 2 empty cells, 2 full cells and, finally, 4 empty cells, the process is as shown in table 2A. For clarity the animations of full cells are shaded. The first column is the fill probabilities due to the background load calculated explicitly. This is then operated on by the transitions appropriate to an additional full cell arriving. This is followed by the two empty cells and so on. The derived parameter Z^+ is shown where it is calculated for the empty extra cell case.

Slot→	1	2	3	4	5	6	7	8	9
Add.Patt→	Full	Empty	Empty	Full	Full	Empty	Empty	Empty	Empty
Initial ↓									
8.89E-01	5.00E-01	6.87E-01	7.75E-01	4.36E-01	2.45E-01	4.44E-01	5.84E-01	6.80E-01	7.45E-01
9.88E-02	3.89E-01	2.33E-01	1.68E-01	3.85E-01	3.80E-01	2.98E-01	2.35E-01	1.92E-01	1.62E-01
1.10E-02	9.88E-02	6.75E-02	4.60E-02	1.37E-01	2.49E-01	1.70E-01	1.19E-01	8.47E-02	6.16E-02
1.22E-03	1.10E-02	1.10E-02	9.02E-03	3.28E-02	9.41E-02	6.48E-02	4.51E-02	3.17E-02	2.25E-02
1.35E-04	1.22E-03	1.22E-03	1.22E-03	6.95E-03	2.48E-02	1.82E-02	1.31E-02	9.48E-03	6.85E-03
1.50E-05	1.35E-04	1.35E-04	1.35E-04	1.10E-03	5.27E-03	4.05E-03	3.08E-03	2.33E-03	1.74E-03
1.67E-06	1.50E-05	1.50E-05	1.50E-05	1.35E-04	9.22E-04	7.66E-04	6.22E-04	4.96E-04	3.91E-04
$Z^+ \rightarrow$	1.66E-06	1.67E-06			1.61E-04	1.45E-04	1.25E-04	1.06E-04	

Table 2A Initial Animation of Background + Pattern

The final column indicates the fill probabilities after the pattern has passed through. These probabilities may be substituted as a new set of initial conditions and the process repeated. Eventually (after 10 cycles in this case) the initial and final states are the same, as shown in table 2B.

Slot→	1	2	3	4	5	6	7	8	9
Add.Patt→	Full	Empty	Empty	Full	Full	Empty	Empty	Empty	Empty
Initial ↓									
6.85E-01	3.86E-01	5.59E-01	6.60E-01	3.71E-01	2.09E-01	3.86E-01	5.18E-01	6.15E-01	6.85E-01
1.68E-01	3.52E-01	2.40E-01	1.89E-01	3.53E-01	3.38E-01	2.78E-01	2.29E-01	1.94E-01	1.68E-01
7.84E-02	1.50E-01	1.13E-01	8.46E-02	1.60E-01	2.45E-01	1.80E-01	1.34E-01	1.02E-01	7.84E-02
3.79E-02	6.12E-02	4.86E-02	3.77E-02	6.47E-02	1.18E-01	8.81E-02	6.62E-02	4.99E-02	3.79E-02
1.73E-02	2.88E-02	2.21E-02	1.70E-02	2.90E-02	5.05E-02	3.89E-02	2.97E-02	2.27E-02	1.73E-02
7.76E-03	1.32E-02	1.01E-02	7.73E-03	1.31E-02	2.23E-02	1.72E-02	1.32E-02	1.01E-02	7.77E-03
3.47E-03	5.94E-03	4.56E-03	3.50E-03	5.93E-03	1.01E-02	7.71E-03	5.92E-03	4.53E-03	3.47E-03
Z ⁺ →	2.67E-03	2.06E-03			4.54E-03	3.47E-03	2.65E-03	2.03E-03	

Table 2B Animation to Stability of Background + Pattern

We have now reached a stable set of state probabilities. A more graphical illustration of the process can be seen in figure 5. This shows how the state probabilities vary during the iterative process, starting from the initial condition until stability is reached.

State	Probability
0	4.88E-01
1	2.60E-01
2	1.39E-01
3	6.36E-02
4	2.85E-02
5	1.27E-02
6	5.74E-03

Table 3 Mean State Probabilities

All that is now necessary is to calculate the mean probability for each state by taking the average value of each row in the stable state in table 2B; of course the initial and final conditions should only be included once. The result is shown in table 3. In order to estimate the equivalent bandwidth it is only necessary to find the random load which approximates to these buffer fill probabilities. Any excess over 0.5 (the background load) will be the equivalent bandwidth.

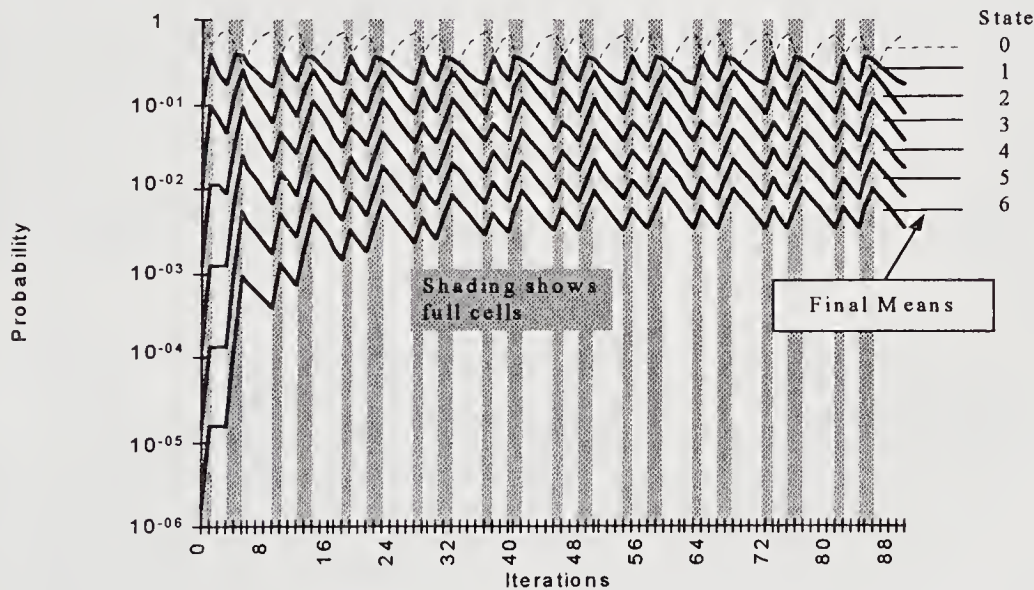


Figure 5 Animation to Stability - Background + Pattern

7. PRACTICAL RESULTS

Let us have a look at results from the MAP mentioned right at the beginning (i.e. 10 full, 25 empty, 11 full, 26 empty, 11 full, 26 empty, 5 full and 133 empty cells). Note that in this case the large number of final empty cells in the pattern will mean that the queue length probabilities will have returned to the level of the background load and a second cycle will not be needed. In fact with other patterns we considered ending with several thousand empty cells, steps were taken to discontinue the animation once the probabilities had reached a stable state.

The tabulation of the resulting probability distribution shows that the buffer fill of 28 occurs with probability of about 10^{-8} . From Figure 6 this is equivalent to a Poisson load just over 0.7. More careful examination comes up with a figure of 0.714. Figure 7 shows the queue length probability distribution due to a Poisson load of 0.5 together with the distribution from the animation with the extra stream.

It can be seen that the distribution due to the combined load is approximately that from a purely Poisson load. Thus it can be argued that the additional pattern contributes a load equivalent to a Poisson load of $0.714 - 0.5 = 0.214$, representing the equivalent bandwidth.

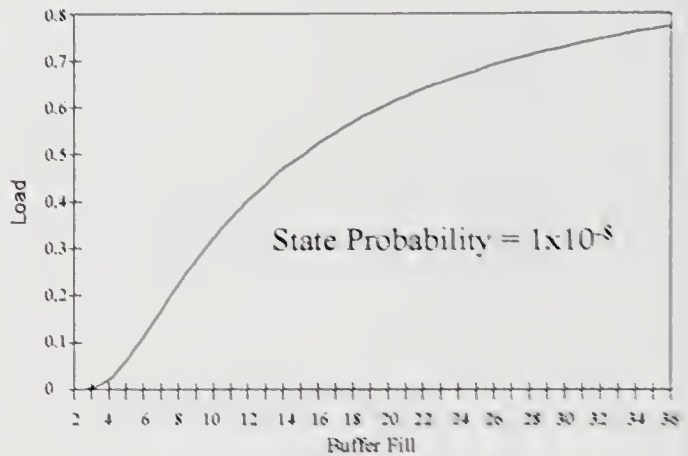


Figure 6 Poisson Load

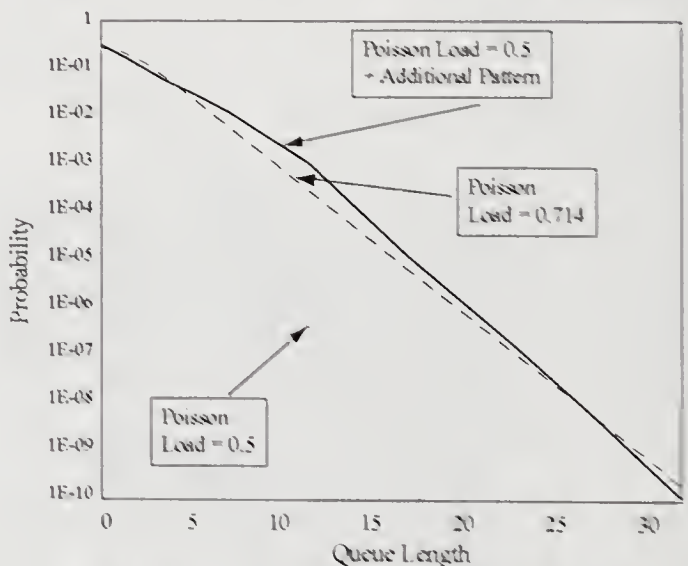


Figure 7 Comparison of Probabilities

8. EQUIVALENT BANDWIDTH: SENSITIVITY TO DERIVATION

In deriving a figure for the equivalent bandwidth two plausible but arbitrary parameters were included. They were the background level of 0.5 and the comparison value of 10^{-8} . Inspection of figure 7 shows that had a higher comparison value been chosen then a higher value for the equivalent bandwidth would have resulted. A summary of the variation for different values of the two parameters is shown in Figure 8 using the

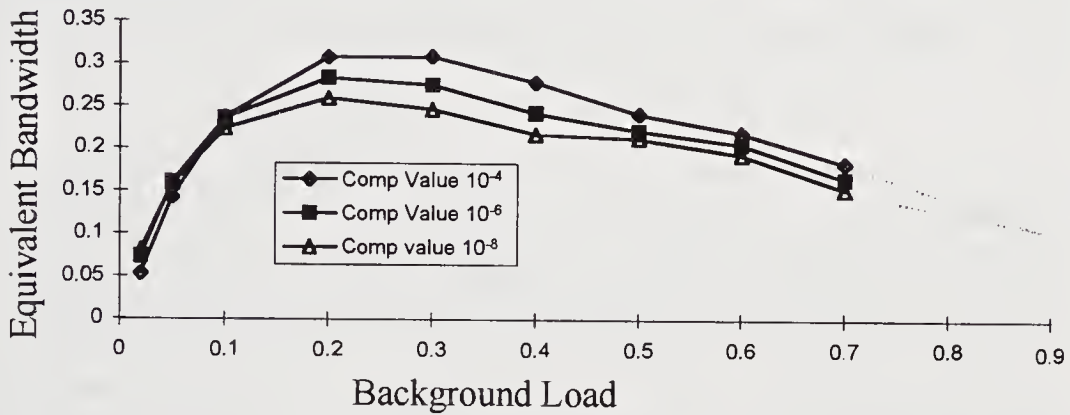


Figure 8 Equivalent Bandwidth: derivation sensitivity

MAP mentioned above. It can be seen that higher values of the comparison value lead to higher estimates of the equivalent bandwidth, implying that the probability distribution function continues to be of the form shown in figure 7. At higher values of the background load the apparent equivalent bandwidth falls. This is inevitable as, in the limit, queues become very large, virtually all cells are full, and the details of the distribution are of little consequence. As the combined load must be less than 1 and the pattern in question contributes 0.1 the trend is indicated by the dotted lines extending to the right. At low background levels the queuing tends to that due to the MAP alone and, as this is a single source, there will be no queuing and hence zero equivalent bandwidth. It can be seen that to ascribe an exact figure to the equivalent bandwidth derived by these means is not possible. However working in the chosen region results in a figure that gives a good indication of the impact of the pattern on the network buffers.

9. PATTERN CELL POSITION PROBABILITY DISTRIBUTION

Let us once again revert to the simple example shown in Table 2B. The columns indicate the queue length probabilities but there is no indication where in the queue the pattern cell might be. If for some reason the pattern cell is always served after the background cells then the first pattern cell will have to wait for an extra time slot before it is served with a probability (from Table 2B) of 0.352, and for 2 extra slots with probability 0.150 and so on. Similarly the other pattern cells will be delayed in the same way. There was always a probability that there was no background cell and the

pattern cell was the only one in the queue in which case it would be served immediately; this arises with probability 0.386. The whole process can be tabulated as shown in Table 3.

Slot =	1	2	3	4	5	6	7	8	9
1st cell	3.86E-01	3.52E-01	1.50E-01	6.12E-02	2.88E-02	1.32E-02	5.94E-03	0	0
2nd cell	5.93E-03	0	0	3.71E-01	3.53E-01	1.60E-01	6.47E-02	2.90E-02	1.31E-02
3rd cell	2.23E-02	1.01E-02	0	0	2.09E-01	3.38E-01	2.45E-01	1.18E-01	5.05E-02
Total	4.14E-01	3.62E-01	1.50E-01	4.32E-01	5.91E-01	5.11E-01	3.16E-01	1.47E-01	6.36E-02

Table 3 Cell Delay Distribution - Pattern cell served last

The table shows that the original pattern cell that had a probability of 1 of being in a particular slot may now occur in one of several slots with appropriate probabilities. The shading indicates the position of the original pattern cells. This is a sort of convolution but the delay probability distribution varies from cell to cell. As the sequence is repetitive probabilities wrap round to the start. The bottom row shows the probability of there being a pattern cell in any slot. Figure 9 illustrates the effect graphically; the clear columns are the original pattern cell positions and the shaded area represents the pattern cell position probability. The different delay variations for the different cells can clearly be seen.

It may be argued that the assumption that the pattern cells are served last is unrealistic. The converse assumption that the pattern cell is served before any background cells arriving in that slot is also easy to evaluate. In that case the first pattern cell will be confronted by the queue length probability distribution due to the cells that have arrived previously. Thus the first pattern cell will not be delayed with probability 0.685, be delayed by one slot with probability 0.168 and so on. Table 4 gives the corresponding figures.

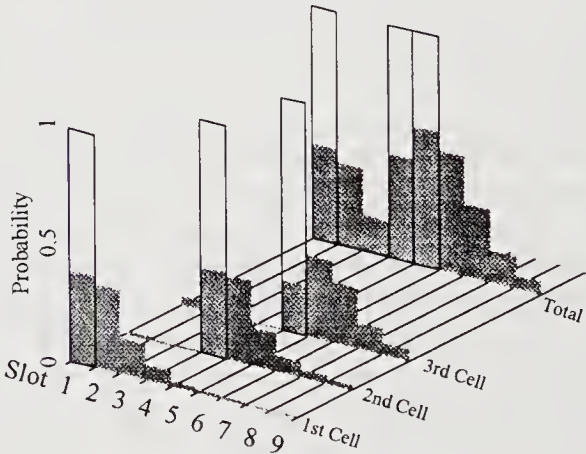


Figure 9 Cell Probabilities

Slot=	1	2	3	4	5	6	7	8	9
1st cell	6.85E-01	1.68E-01	7.84E-02	3.79E-02	1.73E-02	7.76E-03	3.47E-03	0	0
2nd cell	3.50E-03	0	0	6.60E-01	1.89E-01	8.46E-02	3.77E-02	1.70E-02	7.73E-03
3rd cell	1.31E-02	5.93E-03	0	0	3.71E-01	3.53E-01	1.60E-01	6.47E-02	2.90E-02
Total	7.01E-01	1.73E-01	7.84E-02	6.98E-01	5.77E-01	4.45E-01	2.02E-01	8.17E-02	3.67E-02

Table 4 Cell Delay Distribution - Pattern cell served first

The more realistic case of the pattern cell being served at random amongst the background cells is more complex. With N background cells arriving there are $N+1$ places that a pattern cell might find itself and all are equally likely. Hence if N background cells arrive with probability p_N , then the probability that exactly n background cells precede a pattern cell is $p_N/(N+1)$ $0 \leq n \leq N$, Hence the total probability of exactly n background cells before the pattern cell, P_n , is found by

$$\text{summing for all } N. \text{ Hence } P_n = \sum_{N=n}^{\infty} \frac{1}{N+1} P_N$$

Taking the simple case we have been pursuing, and the figures from section 3:

$$P_0 = p_0 + p_1/2 + p_2/3 = 0.7708$$

$$P_1 = p_1/2 + p_2/3 = 0.2083$$

$$P_2 = p_2/3 = 0.0208$$

These probabilities may then be applied in the manner of Figure 4 to the queuing probabilities before the Pattern Cell arrival to give the queue length as seen by the arriving pattern cell. The results are shown in Table 5.

initial	→	Slot 3	→	Slot 4	→
6.85E-01	5.28E-01	6.60E-01	5.09E-01	3.71E-01	2.86E-01
1.68E-01	2.72E-01	1.89E-01	2.83E-01	3.53E-01	3.49E-01
7.84E-02	1.10E-01	8.46E-02	1.18E-01	1.60E-01	2.05E-01
3.79E-02	4.90E-02	3.77E-02	5.06E-02	6.47E-02	9.05E-02
1.73E-02	2.29E-02	1.70E-02	2.27E-02	2.90E-02	3.92E-02
7.76E-03	1.04E-02	7.73E-03	1.03E-02	1.31E-02	1.75E-02
3.47E-03	4.65E-03	3.50E-03	4.66E-03	5.93E-03	7.90E-03

Table 5 Figures of table 2B Modified to Case where Pattern Cell is Queued at Random

These may be rearranged in the same manners as Tables 3 and 4 to get the Cell delay distributions. Figure 10 compares the results of the 3 environments. The tall columns show the original positions of the 3 cells. The other columns show the Pattern Cell position probabilities for the first in the queue, last in the queue and random position environment. Not surprisingly the random environment produces a result between the first and the last in magnitude.

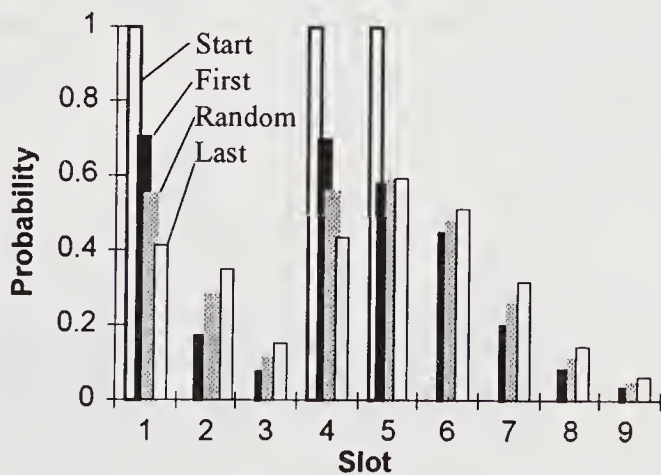


Figure 10 Comparison of Effect of Pattern Cell Priority

10. EQUIVALENT BANDWIDTH OF DISTRIBUTED CELL PROBABILITIES

The technique described in section 6 above assumes that there either 'is' or 'is not' a cell present. Extending this to the environment where a pattern cell is present with a given probability presents no difficulty.

Let c be the probability of a pattern cell. If the probability of n background cells is p_n then the probability of n cells from the pattern and the background together, $pc_n = p_n * (1 - c) + p_{n-1} * c$. It is found easiest to separate the queuing into two processes - arrival and serving. Given a vector representing the first $N + 1$ terms ($s_0 \rightarrow s_N$) of the queue length probability distribution, the first process is to represent

the arrival of cells in the slot: $s'_n = \sum_{i=0}^N s_i pc_{n-i}$.

To avoid the problem of rounding errors the correction process described in section 5 may be applied at this point.

The next process is to represent the serving of the queue. That is to say $s''_0 = s'_0 + s'_1$; for $n = 1 \rightarrow N-1$ $s''_n = s'_{n+1}$; and using the result in section 4, $s''_N = s'^2_N / s'_{N-1}$. This new vector S'' represents the new queue length probability distribution, S . Derivation of mean buffer fill and the equivalent bandwidth may then proceed exactly as described in section 6. A new cell position probability distribution and equivalent bandwidth may also be calculated. If c_m was the relevant cell probability in slot m , with a total number of slots M ($0 \rightarrow M-1$) then the new probability distribution is found:

$$c'_m = \sum_{j=0}^N s_j c_{(m-j)*}$$

The $*$ attached to the $(m-j)$ term is to indicate that if the term becomes negative then M should be added to it to take into account the wrapping effect of the repetitive pattern. If s_j is used then it assumes the pattern cell is last in the queue; s'_j can be used if the first in the queue result is required. If the random case is required then a special

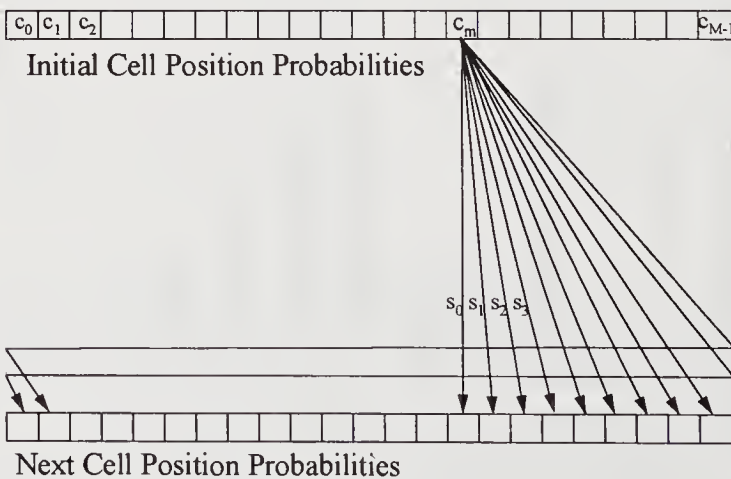


Figure 11 Queuing with Distributed Cell Probabilities

version of s must be calculated after P is calculated as discussed in section 8.

Remembering that s is different for each m in practical calculations it is simplest to follow the practice of Figure 11, accumulating the new cell position probabilities for every value of m .

All results quoted here are based on the 'last in the queue' result. However the 'first in the queue' results are similar except that passage through a queue

has less effect due to the smaller position probability spreading. By repeatedly applying the whole process the cell distribution may be calculated as the pattern of cells passes through a series of queues as shown in Figure 12. From simulation, and also intuitively, it is known that the cells will be distributed about their initial starting position.

One interesting result is that if the cells start fairly clumped initially their equivalent bandwidth will not decrease until they have been through many queues. In fact taking

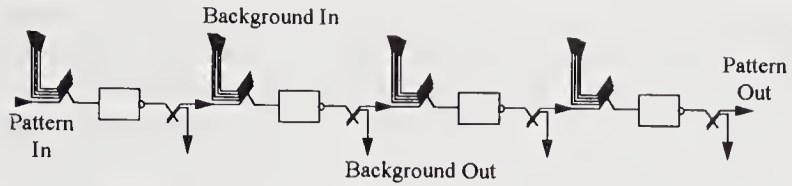


Figure 12 Multiple Queuing Stages

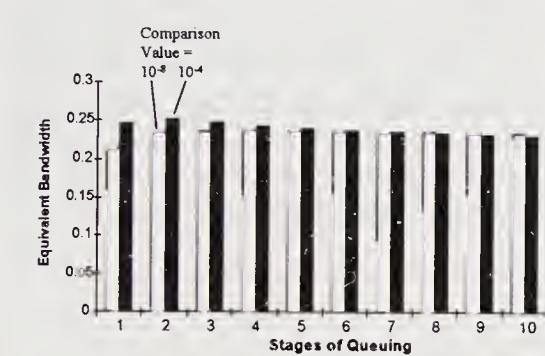


Figure 13 Effect of Several Switching Stages

increases the equivalent bandwidth. With more passages through queues the now smoothed burst of bursts begins to spread giving a reduction in equivalent bandwidth. This qualitative description is quantified in figure 13. The equivalent bandwidth has

the MAP mentioned several times already the effect of passing through the queue is initially to increase the equivalent bandwidth. Several adjacent slots with fractional probabilities of being occupied by a cell have greater bandwidth than an occupied cell surrounded by empty slots during which extra cells may be served. The initial effect of the series of queues is to fill in the short gaps between the short bursts and this

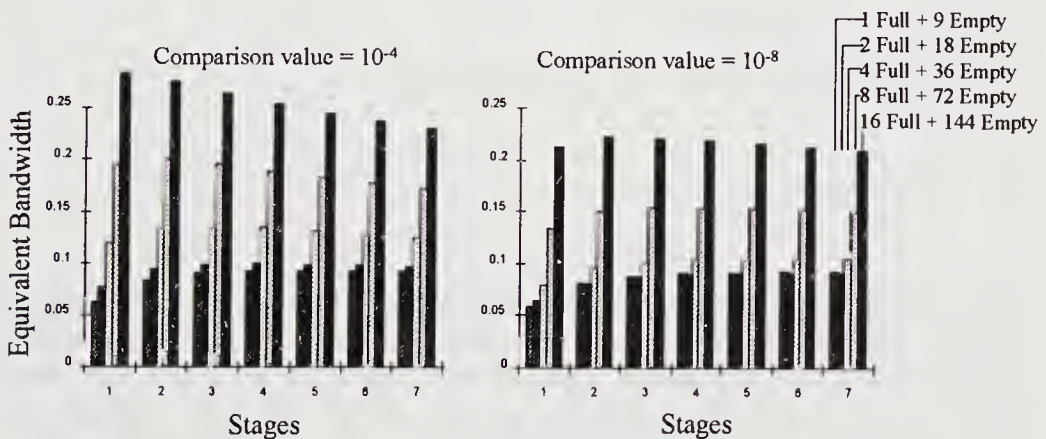


Figure 14 Effect of Bursts Passing Through Several Switching Stages

been calculated against a Poisson background of 0.5 and comparison value of 10^{-4} and 10^{-8} . It is interesting that in this case there is not much difference between the two comparison values; presumably the pattern is "random in nature" and hence does not distort the probability distribution due to the background. Figure 14 also shows the effect of the passage of a bursts of cells, passing through the series of queues. In all cases the mean rate is 0.1 but the cells are in bursts of 1, 2, 4, 8 and 16. The calculation of the equivalent bandwidth has been done at two comparison values - 10^{-4} and 10^{-8} . It can be seen that the fall off in the equivalent bandwidth with passage through several queues is less marked at the lower probability level and that the difference between the two comparison values is much more marked.

11. CONCLUSIONS

It is relatively easy to derive the equivalent bandwidth of a stream consisting of an arbitrary sequence of full and empty cells using the technique of animation described above. This can be extended to the impact of transit through several switching stages. The figure for equivalent bandwidth is not an absolute figure but gives a guide to the impact of a data stream. The value of the equivalent bandwidth, and hence the impact, of the stream is fairly independent of the background parameters and the passage of the stream through the network. Computational requirements are well within the resources of a Personal Computer, with computation times of seconds or minutes.

REFERENCE

- [1] J M Griffiths, L G Cuthbert "Very Selective ATM Statistical Filter" *Electronics Letters*, vol 32, No 7 28 March 96

ACKNOWLEDGEMENT

This work is funded by EPSRC Contract GR/J41635

BIOGRAPHY

JOHN GRIFFITHS graduated from Manchester University in 1966 with a degree in Electrical Engineering. He spent 27 years at BT (formerly the PO) laboratories working on early PCM systems, local network digitalisation, ISDN and, latterly, leading a Division including Submarine Systems, BISDN and LAN interconnection. He has published 2 books: "Local Telecommunications" and "ISDN Explained". He is a Fellow of the IEE, and is a Research Fellow at Queen Mary and Westfield College of the University of London

JONATHAN PITTS graduated from Queen Mary College, University of London, in 1987 with an M.Eng degree in Communications Engineering. Since then he has worked in the Queen Mary College Department of Electronic Engineering on several European RACE projects and, more recently, on UK EPSRC projects concerned with ATM traffic engineering and resource management issues. He completed his Ph.D. on cell-rate accelerated simulation for ATM in 1993, and became a lecturer in the department in 1994.

A scheme for multiplexing ATM sources

J. Naudts, G. De Laet, and X.W. Yin

Universiteit Antwerpen UIA

Departement Fysica, Universiteitsplein 1, B-2610 Antwerpen, België

E-mail: naudts@uia.ua.ac.be, delaet@uia.ua.ac.be, yinxiao@uia.ua.ac.be

Abstract

The introduction of multiple bearer services with different delay characteristics is proposed. In this context statistical multiplexing can be exploited to such an extent that full loading of transmission lines is feasible without cell losses. Strict Usage / Network Parameter Control, based on the Generic Cell Rate Algorithm, is needed. Connection Admission Control can be decided by means of simple arithmetic rules. A switch architecture operating with multiple QoS classes is designed. Simulation results are presented.

Keywords

Bandwidth Allocation, Connection Admission Control, Usage Parameter Control, Network Parameter Control, Generic Cell Rate Algorithm, ATM switch, ATM traffic simulation.

1 INTRODUCTION

In narrowband ISDN 12 different bearer services have been defined (ITU-T, I.200 series); see e.g. (Stallings, 1990), section 6. According to original plans, the introduction of broadband ISDN would have lead to a further increase of this number. The complex situation that would have resulted was avoided by the adoption of Asynchronous Transfer Mode (ATM) which is based on a single bearer service, namely *cell relay*. See e.g. (Händel et al, 1994), chapter 2, or (Minoli et al, 1994), chapter 5. Since then, arguments in favour of multiple bearer services have been formulated (Kröner et al, 1991). Unlike the situation in narrowband ISDN these multiple bearer services would differ only in guaranteed Quality of Service (QoS) (ITU-T, I.356).

Assigning a higher service priority to real-time traffic (such as voice) over non-real traffic (such as data) has been proposed at several occasions. See the introduction of (Lee et al, 1993) and references quoted there. As pointed out in (Kröner et al, 1991), introduction of priorities is not consistent with the idea of the single bearer service. At least two bearer services should be offered to the subscribers, one with high quality of service, one with medium quality. They should be offered either at call or at cell level. The cell loss priority bit in the ATM cell header offers the possibility to introduce two service classes at cell

level. This track has been explored by many authors. The present paper introduces a multiplexing scheme with multiple bearer services at call level. As a side effect of the proposal, some basic problems of ATM technology (statistical multiplexing, queueing in switches and multiplexers, connection admission control, ...) can be solved in an elegant manner.

Overload of a connectionless network leads to degradation of service for all users. In connection oriented networks the setup of new connections is refused when this would lead to congestion. As a consequence, quality of service can be guaranteed to all users. Of course, this requires an accurate knowledge of the conditions leading to congestion. Nowadays there is a strong tendency to relax strict resource management and to replace it by self-regulating mechanisms like discarding cells in case of buffer overflow, flags indicating congestion conditions, traffic regulating tokens, and so on. The alternative followed here is a deterministic network service (Knightly et al, 1995) in which cells are never lost and QoS is guaranteed in a deterministic way. The effect of relaxing conditions, introducing dynamic traffic control mechanisms, can then be studied later on as a small perturbation to a stable and well-balanced system.

The starting point of the present paper is the following observation. In the presence of nothing but Constant Bit Rate (CBR) sources the objectives of

- O1** full load of transmission lines
- O2** no cell losses
- O3** limited cell delays and low cell delay jitter

can easily be met by partitioning the available bandwidth over all sources. The addition of Variable Bit Rate (VBR) sources creates the dilemma of giving up either O1 or O2. Either the sum of all peak rates should add up to at most the total bandwidth with, consequently, a far from optimal line load, or, *statistical multiplexing* is invoked to average out the bursts, resulting in a better use of the available bandwidth and occasional cell losses. The situation studied here adheres to the first option (peak cell rate allocation) but tries to make better use of the bandwidth by filling up holes in the traffic with low priority cells. For this purpose multiple traffic streams with clearly different QoS requirements are needed. In summary, instead of giving up objective O1 or O2, objective O3 is not met for at least part of the traffic (the low priority part).

In this scheme it is essential that the traffic with high priority and small delays is of the VBR type while the traffic used to fill up the holes has constant bandwidth and suffers from rather long delays. CBR traffic with high priority is still feasible. However, it does not lead to any opportunity of using low priority traffic to fill up the capacity of transmission lines. The Available Bit Rate (ABR) service class enters the scheme as a low priority alternative to the CBR service. End-to-end flow control is used to omit the large cell buffers which would otherwise be required at intermediate nodes.

The section on GCRA, shaping, and bursts is used to fix notations and conventions. Next the multiplexing scheme is introduced and rules for resource management are discussed. Priority classes can be organised by cascading several multiplexers. Their use is clarified by means of an example. In section 5 the architecture of a switch which implements priority classes is described. Simulations results confirm the viability of the scheme. In a final section connection admission control is discussed, some considerations are made

about cost effectiveness of the scheme, and possible ways of pricing different services are considered.

2 BURSTY CELL STREAMS

Strict resource management requires a strict enforcement of traffic contracts (ITU-T, I.371; ATM Forum, 1993). Traffic contract conformance is specified by means of the Generic Cell Rate Algorithm (GCRA). A cell stream is said to satisfy GCRA with cell rate r and tolerance τ if the arrival time t_n of the n -th cell is not less than the theoretical arrival time T_n minus the tolerance τ . The theoretical arrival time T_n equals the maximum of T_{n-1} and t_{n-1} incremented with the inverse $1/r$ of the cell rate. In formulas:

$$t_n \geq T_n - \tau \quad (1)$$

$$T_n = \max\{T_{n-1}, t_{n-1}\} + \frac{1}{r} \quad (2)$$

with for the first cell $n = 0$, $T_0 = t_0$. The above version of GCRA is called the virtual scheduling algorithm (an equivalent algorithm is the continuous-state leaky bucket algorithm).

Shaping of cell streams is needed for three reasons. First, the user can shape its ATM source to assure conformance to the traffic contract. Both the network and the user need shaping to remove unwanted burstiness added by the network. Finally, in the estimates about queue lengths an argument involving the maximal length of shaping queues will be used.

Consider a cell stream which satisfies GCRA with parameters r and τ . By means of a queueing buffer the tolerance τ of the cell stream can be reduced to a smaller value τ' . The maximal number of elements in the queue is approximately $r(\tau - \tau')$. The maximal delay of a cell due to buffering is approximately $\tau - \tau'$ (both estimates are only approximate due to the discrete nature of the cell stream).

Bursty cell streams are characterised by specifying two sets of parameters for which they satisfy GCRA (ATM Forum, 1993). The first set is denoted (r_p, τ_p) . r_p is called the Peak Cell Rate (PCR), τ_p the Cell Delay Variation (CDV) tolerance. The other set is denoted (r_s, τ_s) . r_s is called the Sustained Cell Rate (SCR), τ_s the Burst Tolerance (BT). One has $r_p \geq r_s$, $\tau_p \leq \tau_s$, and $r_s \tau_s \geq 1$. Throughout the paper, when not specified, the CDV tolerance τ_p equals the inverse $1/R$ of the cell rate R of the transmission medium. Hence, a bursty cell stream is specified by 3 parameters: r_p , r_s , and τ_s . In what follows it will be called a cell stream with Variable Bit Rate (VBR). If r_s and τ_s are not specified (e.g. because the cell stream is not bursty) then $r_s = r_p$ and $\tau_s = \tau_p$ are assumed. In this case it will be called a cell stream with Constant Bit Rate (CBR) although we do not require that the peak cell rate r_p equals the average cell rate. Hence, in reality the cell rate could be far from constant.

From the estimates quoted above follows that a VBR cell stream with parameters r_p , r_s , and τ_s , can be transformed into a CBR cell stream with parameter r'_p equal to r_s using a queueing buffer of length approximately $r_s \tau_s$. The BT τ_s is often expressed in numbers of cells instead of in seconds. Then the value $r_s \tau_s$ is meant and corresponds (approximately) to the length of the buffer needed to transform the cell stream into a CBR stream.

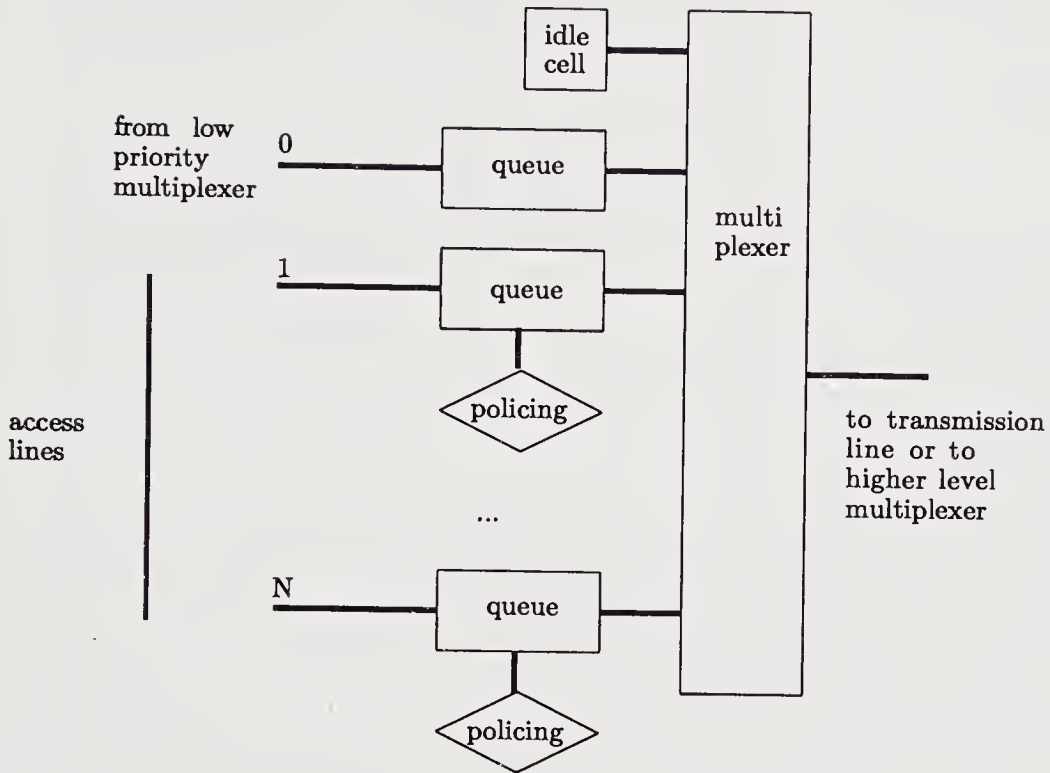


Figure 1 Basic multiplexer scheme.

3 MULTIPLEXER

The multiplexer has a number of identical inputs numbered from 1 to N and one low priority input numbered 0. The high priority inputs are policed to enforce Usage Parameter Control. See Figure 1.

The load of the high priority inputs will be dimensioned in such a way that at most one conforming cell is waiting in each of the N input buffers. The low priority input buffer is served when no high priority cells are present. The strict policing on the high priority inputs together with a correct dimensioning of their usage parameters guarantees that the low priority input receives a specified bandwidth with an upper bound for the delay of its cells. Note that the multiplexer can use a simple round robin algorithm to serve the non-empty high priority queues, although some form of weighted queueing is needed to limit the depth of the input buffers to only one cell.

Let R denote the cell rate of the transmission line. Let $r_p(0), \dots, r_p(N)$ denote the peak

cell rates on each of the inputs. The assumption that at most one conforming cell is waiting for transmission is fulfilled by requiring that

$$\sum_{n=1}^N r_p(n) \leq R. \quad (3)$$

In practice, the inequality is not very strict and can be relaxed somewhat. But then it will happen occasionally that the different high priority inputs hinder each other, and, in this way, acquire extra time delays. The study of this situation is out of the scope of the present paper.

The low priority cell stream is used to fill the holes in the (bursty) high priority traffic. This leads to the second requirement

$$r_p(0) + \sum_{n=1}^N r_s(n) \leq R, \quad (4)$$

where $r_s(n)$ denotes the sustained cell rates of the n -th input. The buffer on input 0 stores low priority cells during bursts of the high priority input channels. If all inputs would be served on equal basis then on each input a buffer of a certain size would be needed to absorb the burstiness of that input. Instead all these buffers are brought together as one large buffer on the low priority input. This is the essential argument used to estimate the size of the buffer on the low priority input.

If no cells may go lost then it is clear that conditions (3, 4) should be satisfied. They still allow full loading of the transmission line. Additional constraints are needed to control delays and buffer sizes. Let $\tau_s(1), \dots, \tau_s(N)$ denote the burst tolerances of the respective inputs. Then one can show that the length of the queue of low priority cells is never larger than

$$\sum_{n=1}^N \tau_s(n) r_s(n), \quad (5)$$

which is the total amount of burst tolerance parameters BT at high priority when expressed in numbers of cells. Hence the delay of a low priority cell is never larger than

$$\frac{1}{r_p(0)} \sum_{n=1}^N \tau_s(n) r_s(n). \quad (6)$$

Sketch of proof. Consider two systems. Both receive exactly the same incoming cell streams characterised by the parameters $(r_p(n), r_s(n), \tau_s(n))$, $n = 0, \dots, N$. Without restriction, assume that the low priority source is CBR. In system I the high priority cell streams are first shaped into CBR cell streams with cell rates equal to $r_s(n)$. As a consequence, only CBR sources arrive at the multiplexer. Because of condition (4) the total cell rate of all sources together is not larger than the available bandwidth. Hence, the traffic can be multiplexed without cell losses and with single cell buffers at every input. System II is the priority system described in the present paper. The service disciplines of the two systems can be coupled. This is done as follows.

- 1 System I uses a weighted queueing discipline.
- 2 If high priority input n of system I is served then also high priority input n of system II is served.
- 3 If low priority input 0 of system I is served then an arbitrary high priority input of system II is served, at least if one can be found which has a non-empty queue. Only if none is found then also the low priority input of system II is served.
- 4 If in system I no cell is ready for transmission then in system II an arbitrary high priority input is served, if one can be found which has a non-empty queue.

Clearly, system II has a better throughput than system I, because of rule 4. Hence it needs at most the same amount of buffering as system I. Assume that in system II a high priority cell is served while in system I the low priority input is served. Then the arriving cell causes an increase in length of the shaper queue at the high priority input of system I while in system II the queue of the low priority input increases by one relative to the same queue of system I. This shows that under rule 3 there is a one-to-one coupling of the queue lengths of the shapers in system I and of the low priority input in system II. Under rule 4 system II becomes more efficient than system I. Both rules together imply inequality 5. \square

Note that (6) implies that the delay is bounded above by

$$\frac{R - r_p(0)}{r_p(0)} \max\{\tau_s(1), \dots, \tau_s(N)\}. \quad (7)$$

Several multiplexers may be cascaded by connecting the output of one multiplexer to the low priority input of the next. See Figure 2. In this way inputs are divided into classes of different priority and correspondingly, different maximal burst tolerances and maximal cell delay jitter. Roughly, the maximal delay jitter in one class of inputs equals the maximal burst tolerance of the class which has one level higher priority, multiplied with the ratio of SCR at high priority / PCR at low priority (both at the higher level — see eq. 7). The amount of traffic in one class of inputs determines the allowable difference between peak cell rates and sustained cell rates in the class of higher priority.

4 EXAMPLES

As an example let us consider a transmission line with a cell rate of $R = 353,000$ cells/sec. Four service classes are provided with characteristics as found in Table 1.

Extreme priority is reserved for a limited number of connections with low SCR, e.g. $r_s = 20$ cells/sec, but high PCR (1,000 or more). With such a connection about ten cells can be issued at PCR with the knowledge that they will receive absolute priority throughout the network. Urgent messages have interesting applications. Service messages such as flow control messages for ABR service, call setup cells, and routing information, need this kind of treatment. But also user applications, e.g. in a client-server context, can profit from urgent short messages. The maximal burst tolerance of the high priority class has been chosen in function of video conferencing. E.g., with usage parameters $r_p = 20,000$ cells/sec, $r_s = 8,000$ cells/sec, and $\tau_s = 0.2$ sec, a burst at peak rate consists

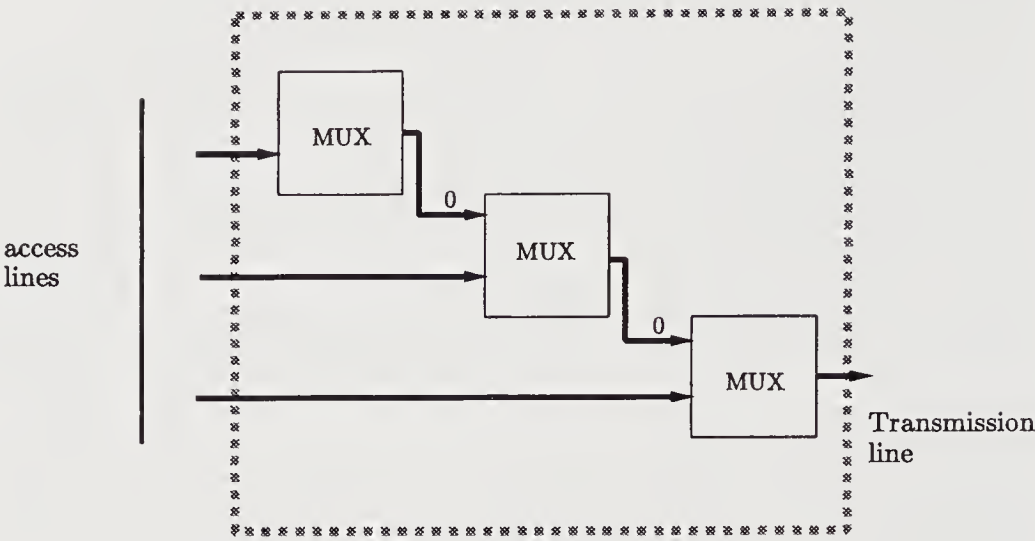


Figure 2 Cascading multiplexers.

Table 1 Example of service classes (times in sec)

	<i>priority</i>	<i>cell delay jitter (sec)</i>	<i>maximal burst tolerance (sec)</i>
(E)	extreme	0.000 5	0.4
(H)	high	0.005	0.2
(M)	medium	0.3	10
(L)	low	5	—

of 2,666 cells and takes 0.133 sec. The maximal cell delay jitter of 5 msec is about what is acceptable for phone calls. In the class with medium priority the maximal burst tolerance is limited to 10 sec. An example of bursty traffic which needs this kind of burst tolerance is interconnection of Local Area Networks (LAN's). In principle, the low priority class should carry CBR traffic because nothing can be gained by allowing VBR. However, this service class is well suited for organising a low cost ABR service. It has a guaranteed bandwidth equal to part of the bandwidth not allocated for services of higher priority. It has also a guaranteed worst case delay jitter.

Table 2 Example of line load

<i>priority level</i>	<i>number of connections</i>	<i>usage parameters</i>	<i>service</i>	<i>comments</i>
extreme	100	(3 000, 20, 0.4)	VBR	control lines
		available bandwidth $R_0 = 353,107$ cells/sec		
		sum of PCR's: 300,000 cells/sec (53,107 not used)		
		sum of SCR's: 2,000 cells/sec (remains 351,107 cells/sec)		
high	1	(79 650)	CBR	450 phone calls
	8	(20 000, 8 000, 0.2)	VBR	real time video channels
	22	(5 000, 2 000, 0.2)	VBR	video conference calls
		available bandwidth $R_1 = 351,107$ cells/sec		
		sum of PCR's: 349,650 cells/sec (1,457 not used)		
		sum of SCR's: 187,650 cells/sec (remains 163,457 cells/sec)		
medium	1	(50 000)	CBR	10 virtual leased lines
	4	(25 000, 5 000, 10)	VBR	network interconnects
		available bandwidth $R_2 = 163,457$ cells/sec		
		sum of PCR's: 150,000 cells/sec (13,457 not used)		
		sum of SCR's: 70,000 cells/sec (remains 93,457 cells/sec)		
low	1	(88 000)	CBR	data channel
	1	(2 000)	CBR	test channel
		available bandwidth $R_3 = 93,457$ cells/sec		
		sum of PCR's: 90,000 cells/sec (3,457 not used)		

The multiplexer requires two small and two large buffers. The queue for multiplexing high priority with extreme priority can be kept small, of the order of 1,000 cells, by limiting the total bandwidth assigned to connections with extreme priority and VBR service. The queue for multiplexing the medium priority inputs with the high priority cells contains of the order of 100,000 cells. The queue for the low priority traffic can become much larger. Suppose that the amount of medium priority traffic is limited to 100,000 cells/sec sustained. Even then the length of the queue can increase to 1 million cells. However, because of the involved delay times it can be implemented using mass memory. Alternatively, if flow control is used for the medium and low priority classes then relatively small buffers can suffice.

Table 2 gives a snapshot of a possible loading of the transmission line. In the table, CBR sources are characterised by a single cell rate, VBR sources by a triple (PCR, SCR, BT).

A second example multiplexing 16 inputs is given in Table 3. It is much less balanced than the previous example. There is important traffic at extreme priority which causes long delays for all traffic of lower priority. One cannot expect this traffic to pass multiple switches while still meeting the goals of Table 1. Still, simulation results reported below show that the behaviour of the network remains predictable. Note that one of the low priority channels is VBR instead of CBR to reduce the nominal load from 100% to 97.17%.

Table 3 Second example of line load

priority level	number of connections	usage parameters	service	comments
extreme	1	(25 000)	CBR	5 virtual leased lines
	1	(150 000, 75 000, 0.006 667)	VBR	?
	available bandwidth $R_0 = 353,107$ cells/sec			
	sum of PCR's: 175,000 cells/sec (178,107 not used)			
	sum of SCR's: 100,000 cells/sec (remains 253,107 cells/sec)			
high	1	(50 000)	CBR	10 virtual leased lines
	1	(103 107, 50 000, 0.01)	VBR	?
	available bandwidth $R_1 = 253,107$ cells/sec			
	sum of PCR's: 153,107 cells/sec (100,000 not used)			
	sum of SCR's: 100,000 cells/sec (remains 153,107 cells/sec)			
medium	1	(121 107)	CBR	?
	5	(5 000, 1 000, 0.1)	VBR	?
	available bandwidth $R_2 = 153,107$ cells/sec			
	sum of PCR's: 146,107 cells/sec (7,000 not used)			
	sum of SCR's: 126,107 cells/sec (remains 27,000 cells/sec)			
low	1	(2 000)	CBR	test channel
	5	(5 000, 3 000, 0.166 667)	VBR	?
	available bandwidth $R_3 = 27,000$ cells/sec			
	sum of PCR's: 27,000 cells/sec (everything used)			
	sum of SCR's: 17,000 cells/sec (remains 10,000 cells/sec)			

5 SWITCH ARCHITECTURE

The overall architecture of the switch is shown in Figure 3. Its components are input cards, schedulers, switching fabrics, and output cards. Both input and output buffering are used. Inside the switching fabrics the buffering is kept minimal.

5.1 Input cards

The traffic arriving at input card i is immediately decomposed according to priority class α . See Figure 4. Next it passes a shaper which limits the peak cell rate of the class as a whole to R_α^i which equals the total bandwidth minus the nominal cell rates allocated for classes of priority higher than α . This is needed while 1) efficient loading of a transmission line introduces extra bursts in the low priority traffic; 2) the high priority traffic has to be protected against bursts of low priority traffic. Consequently, no shaper is provided for the traffic of highest priority. After reshaping, the traffic is further decomposed according to destination d (i.e. number of output card) and stored in small input queues. For convenience, these input queues are labeled (i, α, d) .

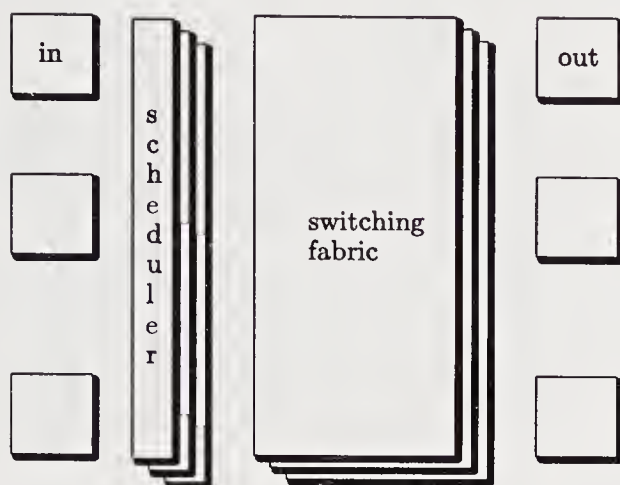


Figure 3 Block diagram of a switch.

5.2 Schedulers and switching fabrics

Schedulers control the dispatch of cells from input buffers to the switching fabric. There is one scheduler for each priority class α and each destination d . It is labeled (α, d) and monitors all input queues (i, α, d) , $i = 1, \dots, N$. Each clock cycle at most one cell with priority α and destination d is given permit to enter the switching fabric. In this way the buffering inside the switching fabric is kept minimal. The schedulers use a round-robin algorithm to select the input queue which obtains permit to transfer a cell to the switching fabric. A weighted queueing algorithm would yield slightly better performance, but was discarded because of the more complex implementation.

There is one switching fabric for each priority class. It has a constant delay and is non-blocking. There are multiple paths (e.g. 2) between each input queue (i, α, d) and the corresponding switching fabric of priority α . One way of implementing the switching fabric could be by means of $\alpha \times d$ busses. Then the schedulers contain nothing more than bus arbitration logic to prevent that several cells are placed on the bus simultaneously.

5.3 Output cards

The cells leaving the switching fabric are fed through a shaper which limits the peak cell rate of the class as a whole to R_α^d which is the total bandwidth of the output minus the

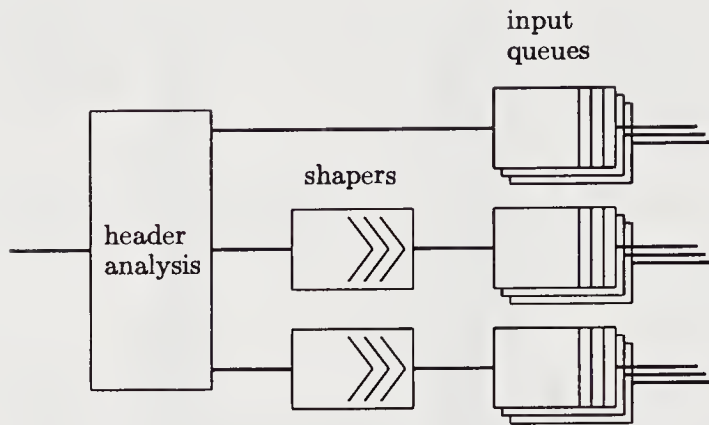


Figure 4 Input Card.

nominal cell rate allocated for classes of higher priority. The output of the shaper feeds the output queues.

The output queues of different priorities are cascaded. See Figure 5. If a shaper with priority α is empty (this implies that no cell of priority α is transferred from the switching fabric to the output card) and the queue of lower priority $\alpha + 1$ is not empty then one cell is promoted from the $\alpha + 1$ -queue to the entry of priority α . Indeed, an empty shaper means that the traffic decreases below the allocated peak cell rate. Then it is time to insert cells of lower priority into this traffic.

The cells of highest priority do not pass through a shaper. The output of the queue of highest priority feeds the transmission line. If this queue (of length one) is empty then a cell is taken from the queue next in priority.

6 SIMULATION RESULTS

We have written a program for numerical simulation of a switch with architecture as described above. Both CBR and VBR sources are simulated. The VBR sources are of a stochastic nature and attain seldom their nominal peak and sustained rates. As a consequence, in all our simulations the observed load of the transmission lines is somewhat lower than the nominal load. We have started by simulating a single multiplexer in order to verify that the principles for loading transmission lines are correct and of practical use. In a second type of experiment we have routed the output of two heavily loaded multiplexers to the inputs of a two by two switch forwarding half of each input to each of the

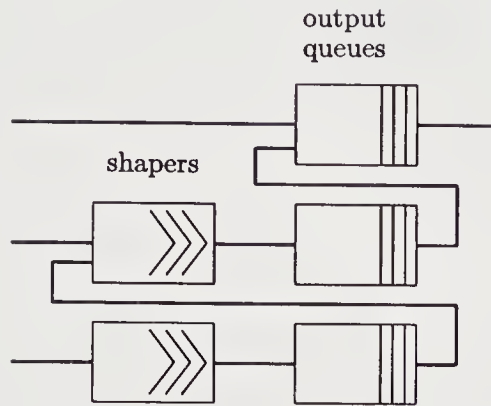


Figure 5 Output Card.

outputs. In this way we could study the effect of feeding a cell stream through multiple subsequent multiplexers/switches. Technically, only one switch is simulated. Each output of the switch can be connected to any input of the same switch in order to realise more complex configurations.

Simulation results for a multiplexer with 137 inputs, loaded as described in Table 2, are found in Table 4. Simulation for 32 sec. takes almost 4 hours of CPU on a DEC Alphaserber 2100 4/275. As expected, no cells are lost when the input and output buffers are dimensioned as predicted by theoretical arguments (see below). Hence the experiment indicates that the Cell Loss Ratio (CLR) is below 10^{-7} . However, from theoretical considerations we expect that it should be identical zero. The average load of the transmission line turns out to be about 92.6%, lower than the nominal load of 99.0%, because the ON/OFF-sources use the allocated capacity in a stochastic manner, not always at maximal rate.

The predictions quoted in Table 4 are calculated as follows. For each VBR channel n in priority class α estimate the number of cells $c_\alpha(n)$ that needs to buy priority from lower class traffic by the tolerance expressed in number of cells, i.e. by $\tau_s(n)r_s(n)$. Note that $c_\alpha(n) = 0$ for CBR connections. The sum $c_\alpha \equiv \sum_{n=1}^N c_\alpha(n)$ is the estimated maximal number of cells that has to be buffered at one stage lower priority. According to formula 7 the quotient $c_\alpha/R_{\alpha+1}$ is the predicted delay jitter for traffic of lower priority $\alpha + 1$. Under the assumption that the main delay for cells of priority α occurs in the output buffer of priority α this is also the estimated maximal delay.

In a second type of experiment, two multiplexers are used to load each of two transmission lines to over 90%. Each multiplexer has 16 inputs and is loaded with the traffic

Table 4 Maximal delays observed in a multiplexer

<i>priority class</i>	<i>maximal delay (sec)</i>	<i>prediction (sec)</i>
extreme	0.000 018	0
high	0.000 11	0.002 3
medium	0.034 4	0.134
low	0.942	2.274

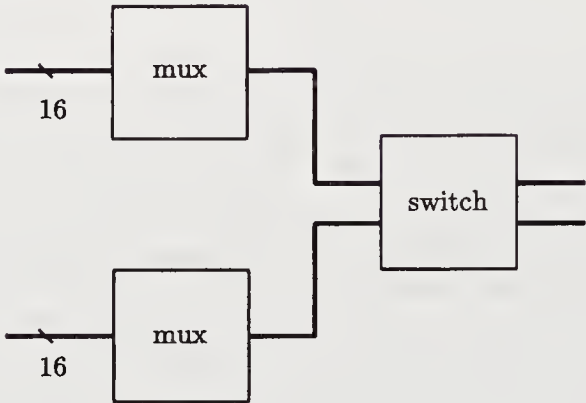


Figure 6 Configuration of the switch.

described in Table 3. The outputs of the multiplexers are fed into a two-by-two switch. The traffic of each input of the switch is about equally divided over each of the outputs. See Figure 6. The input spacers of the switch are now essential to restore the characteristics of the incoming traffic.

The configuration has been simulated for 50 sec. The maximal delays shown in Table

Table 5 Maximal delays observed in the switching experiment

<i>priority class</i>	<i>maximal delay (sec)</i>	<i>prediction (sec)</i>
extreme	0.000 038	0
high	0.004 0	0.004 0
medium	0.010 0	0.010 5
low	0.034	0.047 5

5 correspond with the total time between source and sink. This example shows that the predicted delays can actually be reached. By monitoring the simulated traffic we observe that all connections within the same service class suffer from the same delay jitter. This explains why reshaping on a per class basis suffices.

7 EVALUATION

7.1 Connection Admission Control

How to decide whether a new connection can be added to a partially loaded transmission line? In the first place the availability of enough bandwidth has to be checked. In the present scheme this will depend on the desired maximal delay and hence on the bandwidth still available in the suitable priority class. For a new connection with PCR r_p and SCR r_s the criteria (3) and (4) become

$$r_p + \sum_{n=1}^N r_p(n) \leq R_\alpha \quad (8)$$

$$r_s + r_p(0) + \sum_{n=1}^N r_s(n) \leq R_\alpha \quad (9)$$

The available cell rate in the given priority class is denoted R_α , $\alpha = E, H, M$, or L . For the highest priority class $R_E = R$, for other classes R_α equals R minus the sum of sustained cell rates of all connections with higher priority. For $r_p(0)$ one should use the peak cell rate of lower priority traffic. In practice, $r_p(0)$ can be taken equal to the R_α of the priority class of one lower level. It can be necessary to adapt the values of R_α to make it possible for (9) to be satisfied. In addition, limits can be imposed on the total SCR of one class in order to guarantee specified maximal delays for traffic in classes of lower priority. In this way the bandwidth allocation involves only simple arithmetics.

Two additional properties of the scheme are:

- a CBR connection can always be moved to a class of lower priority with less guarantees on the maximal delays (this is not the case for a VBR connection);
- both PCR r_p and SCR r_s can always be reduced (of course respecting $r_p \geq r_s$); in particular, if a CBR with PCR r_p can be admitted then a VBR with the same PCR but lower SCR can also be admitted (except when not enough queueing memory is available — see below).

In addition to bandwidth allocation it should be checked that there is enough free memory to queue the cells before being transmitted. The solution proposed here is to allocate room for 1 cell per connection plus, in the case of a VBR with parameters r_p , r_s , and τ_s , an additional amount of $r_s \tau_s$ places to be used by lower priority connections. In this way the acceptance of VBR connections does not hinder the further allocation of bandwidth to other connections of possibly lower priority.

Of course, CAC involves many other aspects which are out of the scope of the present paper.

7.2 A cost-effective solution?

Obviously, the proposed multiplexing scheme is only meaningful if the cost of memory is smaller than the cost of transmission. More precisely, let S denote the cost to store one cell in memory during a time equal to the maximal burst tolerance τ_s of a given priority class. Let T denote the cost for transmitting one cell over the transmission line. A necessary condition for the multiplexing scheme to make sense is that S is an order of magnitude smaller than T . This condition seems to be fulfilled on long distance connections.

The billing of a CBR service should be proportional with the cell rate r , say $T \times r$ units per second. For a VBR service the cost of the actual cell transmission is then Tr_s . Two extra contributions have to be taken into account: the excess cell rate $r_p - r_s$ at a fraction λ of the cost T per cell and the cost Sr_s of storage of low priority cells in a buffer of length $r_s\tau_s$. Hence the total cost per second for the VBR service is

$$r_sT + (r_p - r_s)\lambda T + r_s\tau_s S, \quad (10)$$

which can also be written as

$$r_sT \left(1 + \left(\frac{r_p}{r_s} - 1 \right) \lambda + \tau_s \frac{S}{T} \right). \quad (11)$$

The VBR service should be cheaper than a CBR service at peak rate r_p . This leads to the condition

$$\frac{r_p}{r_s} > 1 + \frac{1}{1 - \lambda} \frac{S}{T}, \quad (12)$$

which has useful solutions if $S \ll T$. E.g., with $S = 0.1 \times T$ and $\lambda = 1/3$ a VBR service with $r_p = 2.5 \times r_s$ costs only 1.6 instead of 2.5 times more than a CBR service with the given SCR r_s .

Transmission of lower priority cells costs only $(1 - \lambda)T$ because part of the transmission cost (λT) is paid by corresponding high priority cells in exchange for priority. A further reduction in price can be considered because medium and low priority traffic is used to fill up unused bandwidth.

7.3 Conclusions

Statistical multiplexing of VBR sources poses the non-trivial problem of satisfying simultaneously three objectives: efficient use of bandwidth, no cell losses, and limited cell delay jitter. One solution to this problem is the introduction of multiple bearer services which differ only in quality of service guarantees. The present paper uses theoretical arguments and numerical simulations to show that a multiplexer or switch supporting these multiple bearer services can indeed achieve the three objectives quoted above.

ACKNOWLEDGEMENTS

We thank André De Vleschouwer for critical reading of an earlier version of the paper. We thank Guido Petit for pointing out some of the references.

REFERENCES

- ATM Forum (1993) ATM User-to-Network Interface Specification, v. 3.0.
- R. Händel, M.N. Huber and S. Schröder (1994) *ATM Networks, Concepts, Protocols, Applications*. Addison-Wesley.
- ITU-T (1991) Recommendation I.200 series. ITU-T, Geneva.
- ITU-T (1993) Recommendation I.356, B-ISDN ATM Layer Cell Transfer Performance. ITU-T, Geneva.
- ITU-T (1993) Recommendation I. 371, Traffic Control and Congestion Control in B-ISDN. ITU-T, Geneva.
- E.W. Knightly, D.E. Wrege, J. Liebeherr and H. Zhang (1995) Fundamental Limits and Tradeoffs of Providing Deterministic Guarantees to VBR Video Traffic. *Performance Evaluation Review*, **23**, no 1, 98-107.
- H. Kröner, G. Hébuterne, and P. Boyer (1991) Priority Management in ATM Switching Nodes. *IEEE J Selected Areas Commun*, **9**, no 3, 418-27.
- D.-S. Lee and B. Sengupta (1993) Queueing Analysis of a threshold based priority scheme for ATM networks. *IEEE Trans. Networking*, **1**, no 6, 709-17.
- D. Minoli and M. Vitella (1994) *ATM & Cell Relay Service for Corporate Environments*. McGraw-Hill.
- W. Stallings (1990) *ISDN, An Introduction*. Macmillan.

BIOGRAPHY

Jan Naudts is professor in mathematical physics at the University of Antwerp. Recently, he started research in the domain of telecommunications, with emphasis on mathematical and architectural aspects. Previous research topics include the C*-algebraic approach to quantum mechanics, the statistical physics of orientationally disordered crystals, and random walk in random environment. He obtained a Ph.D. at the Catholic University of Louvain in 1973.

Geert De Laet is a Ph.D. student in the field of telecommunications. He has special interests in the mathematical approach to load balancing and bandwidth allocation in ATM networks, involving topics such as large deviations and Gibbs measures. He obtained a degree in Physics at the University of Antwerp in 1993.

Xiao Wei Yin is working towards his Ph.D. now. He received his M.S. in applied information technology from the Free University of Brussels, Belgium, in 1994. From 1982 to 1991 he worked in the Shanghai Telecommunication Office doing electronic circuit design of telecommunication equipment. He graduated in the field of radio information and engineering from the Shanghai University of Science and Technology in China in 1982.

Shaping of video traffic to optimise QoS and network performance

A. DAGIUKLAS*, M.GHANBARI*, B. J. TYE+

**Department of Electronic Systems Engineering, University of Essex,
Wivenhoe Park, Colchester CO4 3SQ, UK, Tel: 01206-872434,
Fax: 01206-872900, Email:{anasd,ghan}@essex.ac.uk,
+AT&T Bell Laboratory, 77 Science Park Drive, Singapore Science
Park, Singapore 0511, Tel : 065 8706538, Fax : 065 8720140,
Email : bjtye@zpcpbl.att.com*

Abstract

A method of shaping the video traffic within the video encoder is proposed. At the intraframe coded frames, where the maximum number of bits are generated, the coder constrains its generated bit rate through a leaky bucket mechanism. A sliding window is also used to maximise network utilisation without violating any of the traffic parameters declared at the call set-up. The impact of the shaping mechanism on both coding and network performance are studied. It is shown that for video sequences with scene cuts, shaping the video traffic under a certain peak-to-mean ratio optimises both network performance and perceived image quality.

Keywords

Traffic and Congestion Control

1 INTRODUCTION

ITU-T has proposed ATM as the mechanism for multiplexing/switching in the future B-ISDN (ITU-T 1991). A key challenge in the ultimate success of ATM is to define and implement a congestion control strategy that provides an efficient sharing of network resources among different services with diverse traffic characteristics. Such a congestion control comprises of three sections, namely control of access of the customers to network resources, policing the traffic flow of each user and protection of the Quality of Service (QoS) against possible fluctuations of the traffic flow above the channel capacity .

Video services are expected to share a large portion of the traffic handled by ATM networks. A critical aspect of VBR coding and transmission is the real-time constraints for VBR video data. The network has to ensure the on time delivery of data, while on the other side the encoder has to provide the appropriate shaping functions in order to improve the channel performance. This shaping function can be used by the encoder to regulate its traffic at the ingress of the network.

The paper is structured as follows: Part 2 describes the policing functions/(UPC) methods for policing a service in an ATM network. Part 3 presents the UPC methods proposed for regulating video services. Part 4 gives the impact of the proposed scheme on the network performance. Part 5 investigates its impact on the perceived image quality. Finally, conclusions are given in Part 6.

2 POLICING FUNCTION/USAGE PARAMETER CONTROL (UPC)

After the connection is established, the network has to monitor the conformity between the declared and the actual cell stream parameters at the ingress of the network. This is enforced to protect the network resources from possible malicious or erroneous users who may exceed the traffic volume declared at the call set-up and thus overload the network. This function called user parameter control (UPC) is performed at the user network interface/ network network interface (UNI/NNI) for each existing virtual path/virtual circuit (VP/VC), controlling its traffic flow based on the declared traffic parameters. If a VP/VC is detected violating the agreement, its cells can either be discarded or tagged for later discard when congestion arises. The policing function can be characterised by the following attributes (BAE, 1991), (IEEE, 1991), (IEEE, 1992):

I) The UPC mechanism should be selective with respect to the policed parameters. It should be able to distinguish the trade off between traffic fluctuations during normal operation from real traffic violations.

II) It should respond rapidly to parameter violations.

III) The mechanism should be simple and flexible to implement.

Some of the most common policing techniques involve leaky bucket and window mechanisms. Leaky bucket (Niestegge, 1990) is a virtual buffer (bucket) with a constant service time, as is illustrated in Figure 1a. Once the buffer becomes full a violation is detected. The service rate of the virtual buffer corresponds to the rate to be policed (for example, peak bit rate (PBR) or mean bit rate (MBR)) assuring that UPC algorithm tolerates fluctuations caused by cell delay variation (CDV) or burstiness. The size of the bucket is determined by the maximum burst length that the user is allowed to submit to the network. Another implementation of the leaky bucket is to control the traffic flow by a means of tokens (Sidi, 1989). A queuing model for this method is shown in Figure 1b. An arriving cell enters the bucket after it has received a token pool. If no tokens are available, a cell must wait in the queue until a new token is generated. Tokens are generated at a fixed rate corresponding to the bit rate to be policed.

A window is a fixed time interval, defined as a number of time slots in an ATM VP, which is used to measure the number of cells within this time interval (Bae, 1991), (IEEE, 1991), (IEEE, 1992), (Roberts, 1992). There are two versions of window mechanisms, namely jumping window and moving/sliding window, as shown in Figure 2. The jumping window consists of non-overlapping consecutive time intervals that counts the number of cells delivered from a source within the interval. A new interval starts immediately at the end of the preceding interval where the associated counter is reset to zero. In the moving window, the window slides

continuously through the time. Thus, the arrival time of each cell is stored and a counter is incremented by one for each new arrival. Exactly T time units after an arrival of an accepted cell, the counter is decreased by one.

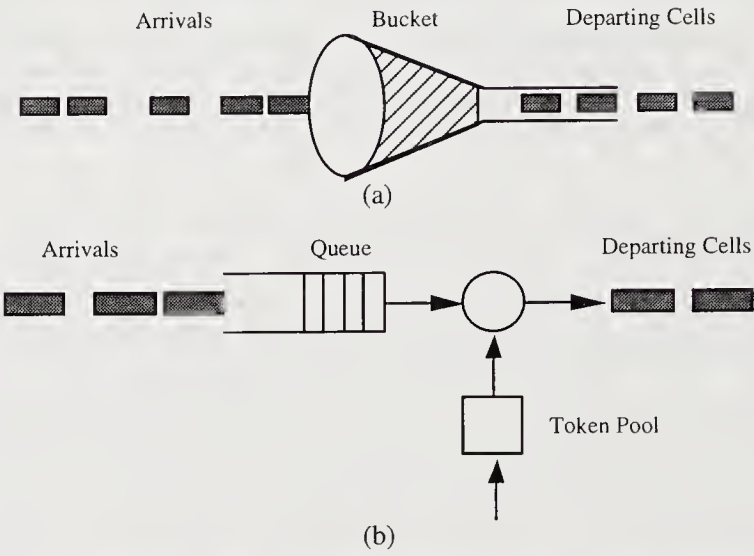


Figure 1: Schematic representation of leaky bucket.

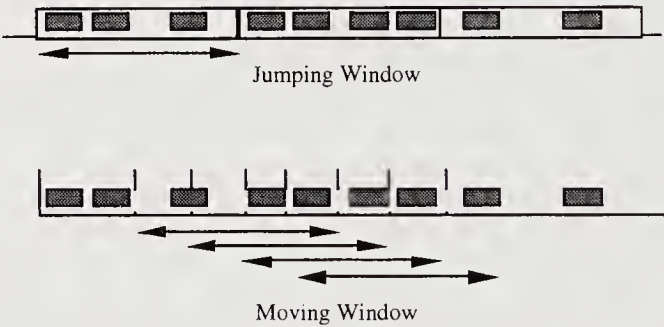


Figure 2: Schematic representation of window mechanisms.

3 EMPLOYMENT OF POLICING MECHANISMS IN VIDEO SERVICES

Much attention has been paid to the implementation of the policing functions in packet video. Such attention stems from the fact that real time services, such as video, prefer to constrain the generated traffic according to the declared MBR and PBR such that no penalty is imposed from

the policing function. Such penalty may lead to deterioration of the QoS due to the discard of cells which are important for the reconstruction of the pictures at the decoder. The policing function is normally imposed at the UNI. Once a violation from a source is detected, cells from that source may be discarded or tagged for later discard. As an option a feedback from the UNI to the source can be set up to regulate the traffic as required.

Ratheb (1993) has studied the impact of the policing functions on video services. His studies show that efficient policing can be achieved by restricting the PBR to a reasonable value. Among the leaky bucket, moving and jumping windows, the leaky bucket exhibits better performance than the window mechanisms. Policing of the MBR seems not to be realistic for either of the mechanisms due to the large bucket/window requirements. This is because observations have shown that cell losses occur in clusters and for a given policed rate, the size of the bucket/window has to be extremely large.

The imposition of UPC can badly damage QoS of those services violating their traffic descriptors. This is more pronounced in an interframe coded video, where loss of one cell may propagate through several video frames. It would be beneficial to the user himself to control his generated bit rate prior to being penalised by the network operator. This is because if the encoder codes pictures at a lower bit rate and image quality is temporarily degraded, since the decoder without cell loss can track the encoder, the picture quality can be improved later. On the other hand if cells are lost due to the network policing, since the decoder can not track the encoder, the picture quality will remain poor for a long time, which is very objectionable.

Harasaki and Yano (1993) have used a leaky bucket to police both PBR and MBR. They have demonstrated that the leaky bucket size should be quite long (possibly as long as several seconds) but not prohibitively long from network designers' point of view, in order to allow constant picture quality for most of the time during a long video program.

Kawashima and Tominaga (1993) have used a sliding window to police the MBR. This method utilises variability of bit rate under the constraints by the UPC and its influence on the QoS. It has been reported that transmission of video under this mechanism shows significantly better image quality than the conventional constant bit rate (CBR) transmission in scene changes. In addition, when the sliding window is small (10 to 30 frames) image quality is very poor in the areas of pictures with zooming or panning. We have adopted a more integrated approach where a shaping mechanism is used to police the declared traffic parameters and adjust the actual bit stream accordingly, such that both network and codec performance is optimised.

4 IMPLEMENTATION OF THE SHAPING MECHANISM

The proposed shaping mechanism imposes two constraints in the generated bit rate. The first constraint deals with the shaping of PBR and the second is concerned with the control of MBR. Thus, the objective of the shaping mechanism is the best usage of the available resources provided by the network operator, while at the same time the user tries not to violate the contract declared at the call set-up. These constraints are described in the following sub-sections.

4.1 Shaping/Smoothing of the PBR

Video codecs for ATM networks are VBR oriented. The bit rate variation is a function of scene content and motion of moving objects. The PBR (the maximum number of bits in one frame period) is normally generated at scene cuts, where the pixels are coded with an intraframe method. In this study, an H.261 standard video codec was used, where images are interframe coded using motion compensation for greater compression. At scene changes, the encoder

switches to an intraframe mode, generating its PBR. The bit rate can be regulated by adjusting the quantiser step size. For example in the reference model simulation coder (RM8, 1989), the quantiser step size can be changed at the start of each group of pictures (GOB) or at one third of them (11 macroblocks) (ITU, 1990). This technique in conjunction with the rate smoothing buffer is employed in circuit switched network applications to deliver constant bit rate video into the channel.

The proposed strategy for the shaping of the PBR is to employ two virtual buffers. The first buffer performs like a leaky bucket and its occupancy is used as a feedback to the encoder to control the quantiser step size. Note that increase (decrease) in the quantiser step size results in the decrease (increase) in the generated bit rate. The second buffer counts the total number of bits generated within the scene cut frame. A threshold value, s , is imposed at the counter to control the quantiser step size further whenever is necessary. The dimensions of both buffers are equal to the PBR declared at the call set-up.

The method employed in RM8, was used to detect a scene cut by comparing the variances of intraframe and interframe coded macroblocks. Then, if in the first few GOBs (e.g. 2-3 GOBs) the majority of the macroblocks are intraframe coded, it can be assumed that the whole frame

will be intraframe coded, i.e. detection of a scene cut. This introduces $\left(\frac{1}{6}, \frac{1}{4}\right)$ of a frame delay, corresponding to almost 6-9 ms in a 30 Hz video. At scene cuts, where the codec switches to the intraframe mode, the quantiser step size is adjusted at the start of the frame. Since the aim of the peak constraint is the reduction of the PBR which occurs at scene cuts, then the quantiser step size has to be increased. It was found experimentally that a good starting point is to set the quantiser step size q_p to $1.5 \times q_{int}$, where q_{int} is the quantiser step size during the interframe coding mode. In our experiments, q_{int} was set to 12. While coding the scene cut frames, the quantiser step size is adjusted every 11 macroblocks based on the fullness of the leaky bucket. The adjustment of the quantiser step size is controlled, based on RM8 where the quantiser step size q_{sc} , is :

$$q_{sc} = 2 \times \text{INT} \left(\frac{32 \times b_i}{b_{max}} \right) + 2 \quad (1)$$

where b_i denotes the leaky bucket fullness after coding each macroblock and b_{max} is the control buffer dimension determined by the targeted PBR. The initial leaky bucket content is calculated from (1), such that quantiser step size is q_{int} . The leaky bucket is filled up with the rate of generated data at each macroblock, but it is emptied at rate $\frac{\text{PBR}}{396}$ at every macroblock (there are 396 macroblocks in each frame). Once the buffer content reaches the threshold s , the quantiser step size is further adjusted by :

$$q_{sc} = q_p \times b_{av} \times \left(\frac{1 - \frac{\text{coded MB}}{\text{total MB}}}{b_{max} - \text{leaky bucket level}} \right) \quad (2)$$

where b_{av} is the target mean bits/frame. It was found, that a threshold of $s = 0.7 \times b_{max}$ is a good indication of the fullness of the virtual buffer that controls the generated bit rate in the scene cut.

4.2 Control of the MBR

The shaping of the PBR itself is insufficient to yield a reliable control mechanism since the other important parameter declared during the call set-up is the MBR. The encoder should employ a method such that MBR is neither underestimated nor overestimated. Underestimation would lead to cell loss while overestimation would be poor utilisation of the network resources that the user has paid for. For this purpose, a sliding window may be employed to monitor the short term MBR which is used as a guideline to estimate the long term MBR. For the target MBR of b_{av} bits/frame, the expected bit rate within the window of size w frames is $w \times b_{av}$. The actual generated bit rate, w_{sum} , within this interval is:

$$w_{sum} = \sum_{i=1}^w f_i \quad (3)$$

where f_i is the generated bit rate at frame i . Thus, at any time instant, the deviation, d_{ev} , of the actual sum from the expected one within the window is:

$$d_{ev} = (w \times b_{av}) - w_{sum} \quad (4)$$

which is used to code the new frame.

To code a new frame, the window is shifted by one frame. The frame which is dropped out of the window with bit rate f_{rem} , is added to the deviation bit rate to estimate the allowable bit rate for coding the new frame f_{new} as:

$$f_{new} = d_{dev} + f_{rem} \quad (5)$$

The quantiser step change Δq for the new frame in the window is calculated by normalising f_{new} to the b_{av} :

$$\Delta q = \frac{b_{av} - f_{new}}{b_{av}} \quad (6)$$

Thus the quantiser step size for the new frame, q , is derived from:

$$q = q_{min} + \Delta q \quad (7)$$

To preserve the characteristics of VBR coding, the upper bound of the quantiser is crucial. If the variation in the quantiser step size is quite large, it may degrade the picture quality. In addition, the overall consistency of picture quality is affected. On the other hand, small variation in the quantiser step size may cause w_{sum} to exceed $w \times b_{av}$. Thus, in order to compromise the above effect, Δq is limited to a maximum of four while the lower bound of q is set to q_{min} . The newly adjusted quantiser step size is used to code the next frame. No further transition in the quantiser step size is allowed within the next frame to obtain consistency within the frame.

5 THE IMPACT OF TRAFFIC SHAPING ON THE DECODED IMAGE QUALITY

A typical video sequence containing several scene cuts was used to evaluate the effect of the traffic shaping on the decoded image quality. A fixed quantisation step size of 12 was used to code the sequence. Figures 3 and 4 illustrate the cell generation and the PSNR profiles respectively for the video sequence under study.

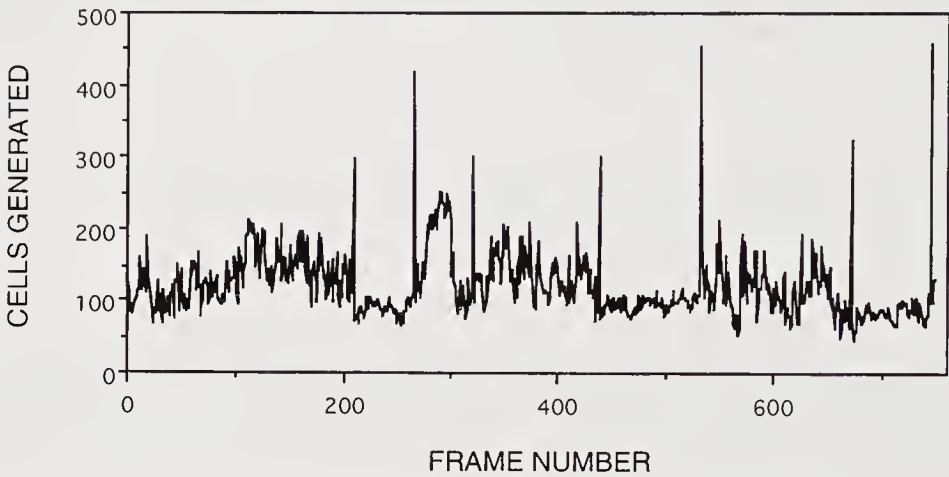


Figure 3: Bit Rate profile of a typical video trace with several scene cuts coded with an H.261 video codec.

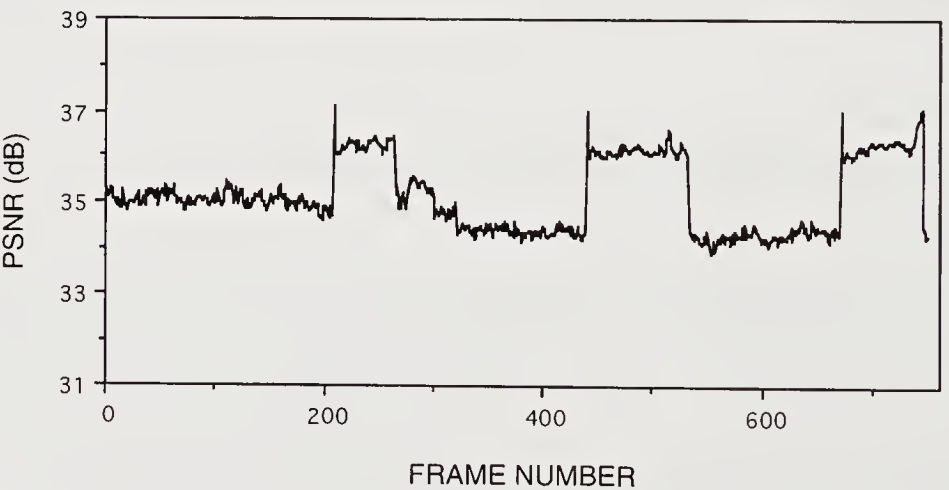


Figure 4: PSNR of a typical video trace with several scene cuts coded with an H.261 video codec.

At a fixed quantiser step size ($q=12$) the picture quality is almost constant. Small quality variation is due to the scene dependency of coded video. The sequence was also coded under the shaping constraints. The impact of the shaping constraints on the bit rate and PSNR is demonstrated below.

5.1 Peak To Mean (P/M) ratio

For a given MBR, the constraint imposed on the PBR reduces P/M. Figure 5 illustrates the cell generation profile of the video trace when P/M is reduced from its unconstrained value (3.5) to 2.5. Due to the PBR constraint, PSNR is degraded at the scene cuts, as illustrated in Figure 6.

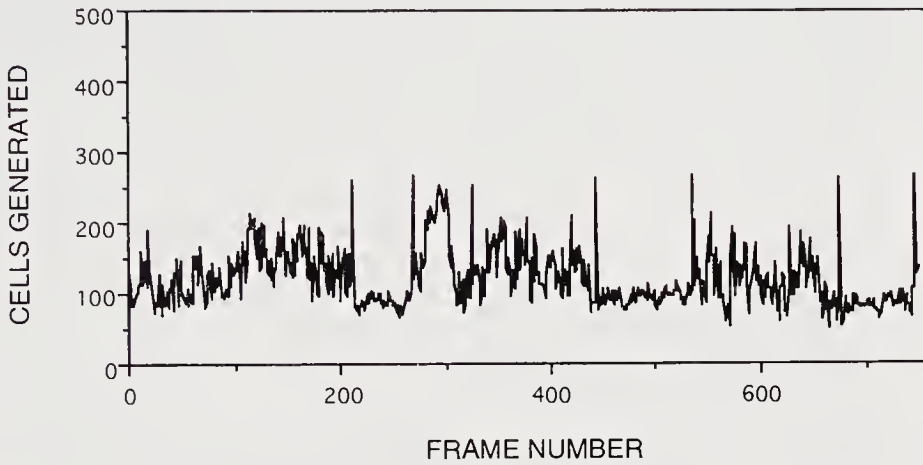


Figure 5: Bit Rate profile of a typical video sequence under the shaping mechanism.

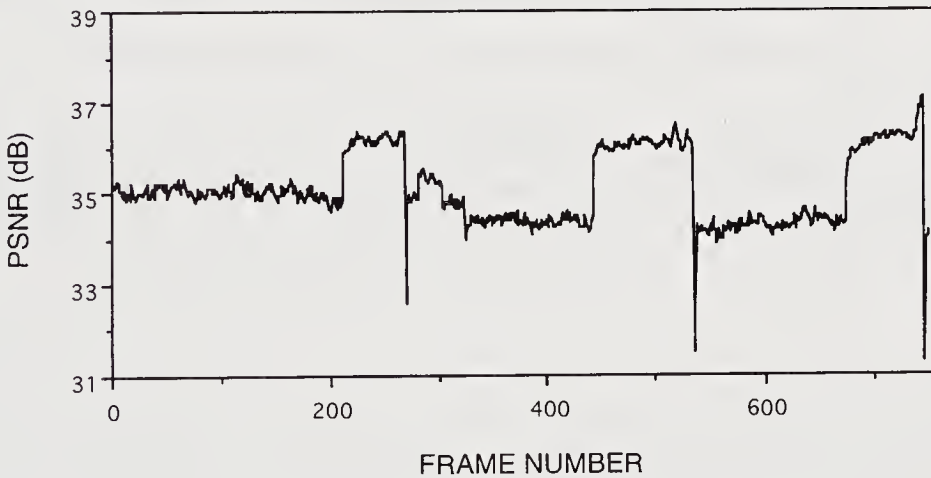


Figure 6: PSNR of a typical video sequence under the shaping mechanism.

The allowed degradation is picture dependent and is subject to the visibility threshold of the observer. The drop in the bit rate at scene cuts, causes the bit rate in the subsequent frames to sustain in a high level until the MBR converges to its long term average. The smaller the P/M (larger constraint imposed in the PBR), the worse is the degradation in the PSNR at scene cuts. Since in normal TV programmes the scene cut frequency is small (1 every 5-9 s (Hughes, 1993)), the constraint imposed on PBR does not alter the MBR significantly.

5.2 Window Size

The window size determines the number of frames used to calculate the short term MBR. It has been suggested (Kawashima, 1993) that a selection of window in the range of 50 to 150 video frames would give good estimation of the MBR while at the same time image quality does not degrade in scenes of panning or zooming. The results have shown that by using different window sizes for a given PBR and MBR, variation in the overall PSNR is not significant, as Figure 7 illustrates.

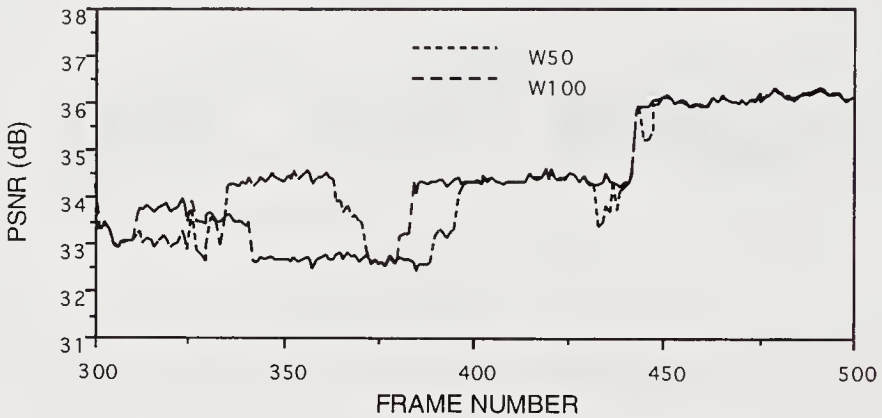


Figure 7: Effect of window size on the PSNR of a typical video under the shaping mechanism.

No constraints are imposed on the bit rate up to the point where the window is full assuming that there are no scene cut frames in this period. When the coder starts controlling the bit rate, the MBR for the overall traffic tends to converge towards the long term mean, as illustrated in Figure 8. In addition, once the user selects an appropriate window size in the range of 50-150 frames, the window size has no impact on the generated bit rate as illustrated in Figure 9.

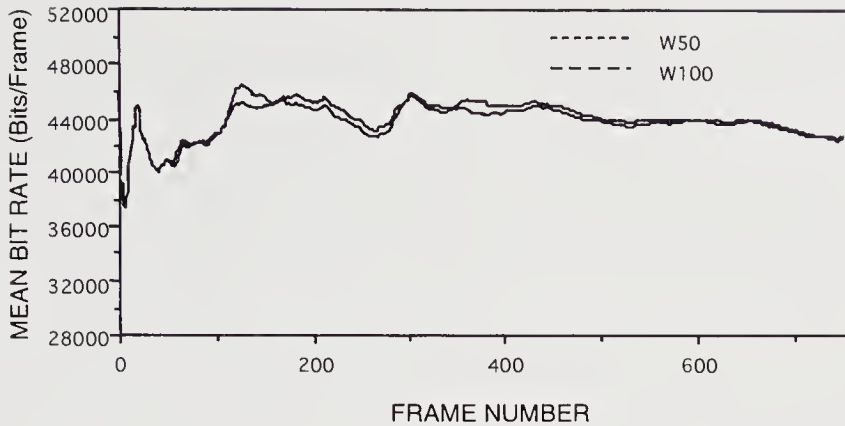


Figure 8: Effect of the window size on the MBR of a typical video under the shaping mechanism.

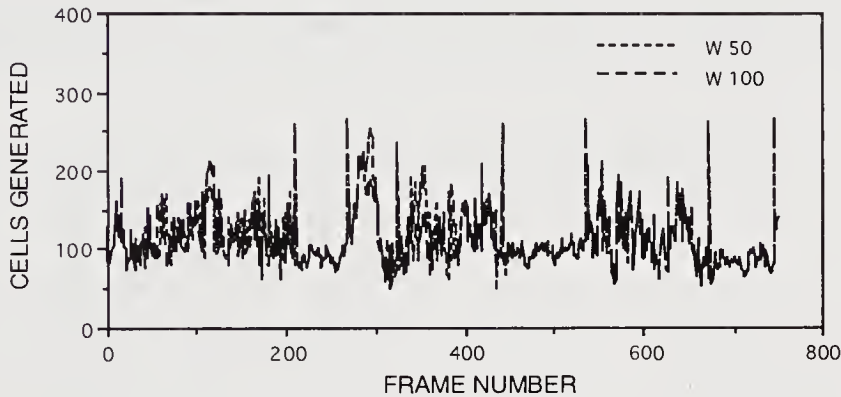


Figure 9: Effect of the window size on the bit rate of a typical video under the shaping mechanism.

6 THE IMPACT OF TRAFFIC SHAPING ON THE NETWORK PERFORMANCE

Although the reduction of the peak bit rate at the intraframe coded frames, leads to poorer PSNR at these frames, it is expected that it will ease network congestion reducing the cell loss rate. To study this improvement, a single multiplex of eight homogeneous video sources was considered. An 8-cell size buffer was used at the input of the multiplex to withstand simultaneous cell arrivals from the eight sources (one cell per channel). A FIFO policy was employed to serve the buffer.

The output of the encoder generates bits per macroblock. Every 44 bytes of video data were packetised into the payload of ATM cells and a list of interarrivals of video cells was generated. Each sequence was considered as a circular linked list of homogeneous video sources. An event driven simulation was adopted for the generation of the traffic of each video source. For each video source a different offset point (randomly selected) in the list was used to ensure that all sources are not identical on a cell by cell basis. The distances between the starting points were taken larger than 10 frames such that correlation between cell generation was made small.

It was observed that reducing the PBR or (P/M) decreases the cell loss rate. This is because lowering P/M reduces the burstiness of the incoming data at the intraframe coded frames and the small multiplexing buffer is less flooded. However at much lower values of P/M, the cell loss rate rises again. This is due to the fact that although image quality at scene cuts is impaired, the interframe errors (due to coding distortions) in the subsequent frames remain high for a few frames, till all the coding errors are cleared. Thus, the limited multiplexed buffer is subject to a flow of data for a longer time. Therefore, there should be an optimum value for P/M, where the cell loss rate is the smallest. Figure 10 illustrates the cell loss rate for various network loadings when P/M is reduced from its unconstrained value of 3.5 to 2.0. The smaller the loading factor, the larger becomes the difference in cell loss ratios for different P/M ratios. For example at 50% network load, the optimum P/M ratio for this sequence is 2.25. For such P/M ratio, the cell loss is less than 1/6 of the unconstrained video. As network load increases, more cells face the full buffer and thus the difference in cell loss ratios decreases.

The optimum value of P/M shown in Figure 10 can also be justified from an investigation of the burstiness of the generated data under various P/M ratios. Here we define burstiness as the number of generated cells per macroblock. Figure 11 illustrates the mean values of burstiness for various P/M ratios, showing that P/M of 2.25 has the least burstiness.

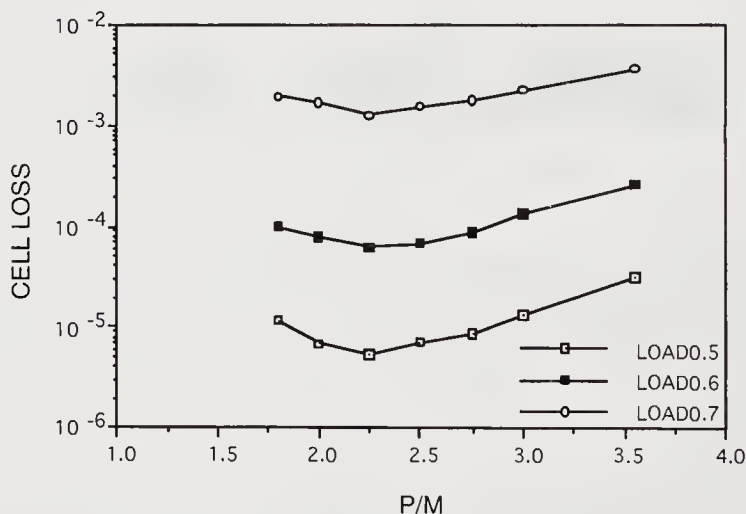


Figure 10: Cell loss rate for different P/M ratios at various network loads.

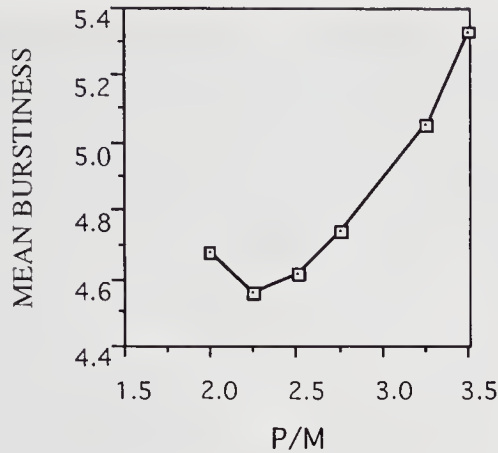


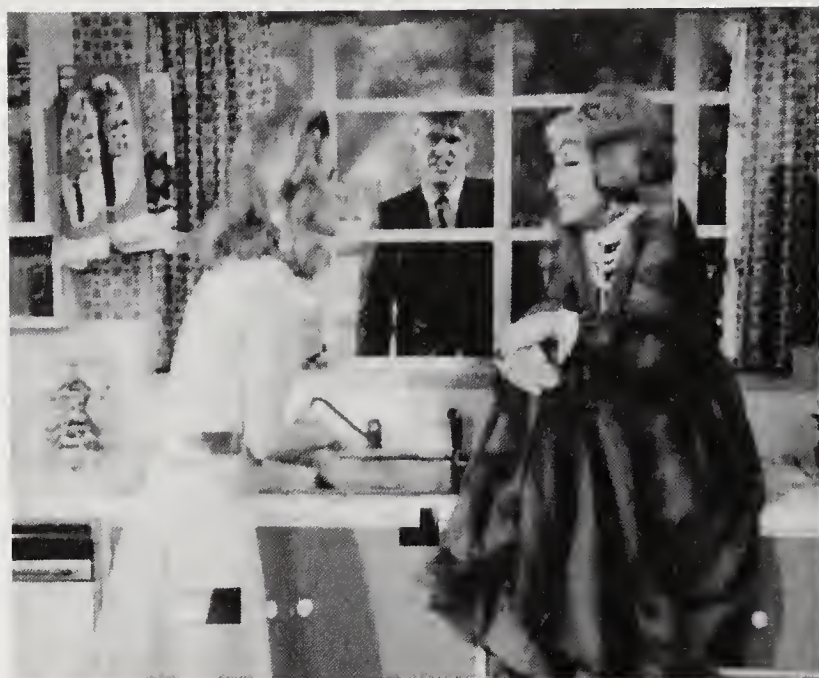
Figure 11: Mean burstiness of the generated video at different P/M ratios.

7 THE IMPACT OF TRAFFIC SHAPING ON THE QoS OF VIDEO SERVICES

The video sequence with the characteristics of Figure 10 was used to evaluate the PSNR of the coded pictures under the uncontrolled and the optimum constrained P/M ratios. As Figure 10 shows, for the uncontrolled P/M value of 3.5 and multiplex buffer size of eight cells, the cell loss rate at network load of 0.5, is almost 4×10^{-4} . This value for the optimum constrained P/M value of 2.25 is nearly 6×10^{-5} , which is about 6 times smaller than that of the uncontrolled case. Assuming that cell loss occurs in clusters and are confined within a frame, then for the 750 frames sequence under study the cell loss rates in a particular frame are almost 2.7×10^{-1} and 4.5×10^{-2} for uncontrolled and optimum constrained P/M ratios respectively. Two cases were examined: cell loss at a scene cut (intraframe coded) frame and cell loss at a interframe coded frame

7.1 Cell Loss in a Scene Cut Frame

Figure 12a illustrates a scene cut picture frame of the sequence under the uncontrolled P/M ratio. The scene cut frame of the sequence was exposed to 27% cell loss. Although picture quality in the non-lossy areas is good, the artefacts due to cell loss are very disturbing. Due to the interframe nature of the codec and the fact that the encoder is unaware of the cell loss, in the decoded images the artefacts will propagate through the image sequences and can last for a long time, as shown in Figures 13a and 14a, which display the picture at one frame and five frames respectively after the lossy scene cut frame. These artefacts can be cleared when the entire frame is updated with intraframe coded information (Ghanbari, 1993). Figure 15 illustrates the propagation effects of cell loss for the sequence under study. At the instant of the cell loss the image quality drops from its nominal value of 36 dB to 26 dB. It may take several frames for the decoder to completely recover from the lost cells of one frame. For example in the H.261 codec,



(a)



(b)

Figure 12 A scene cut frame with a cell loss rate of a) 27% unconstrained and b) 4.5% optimum constrained.

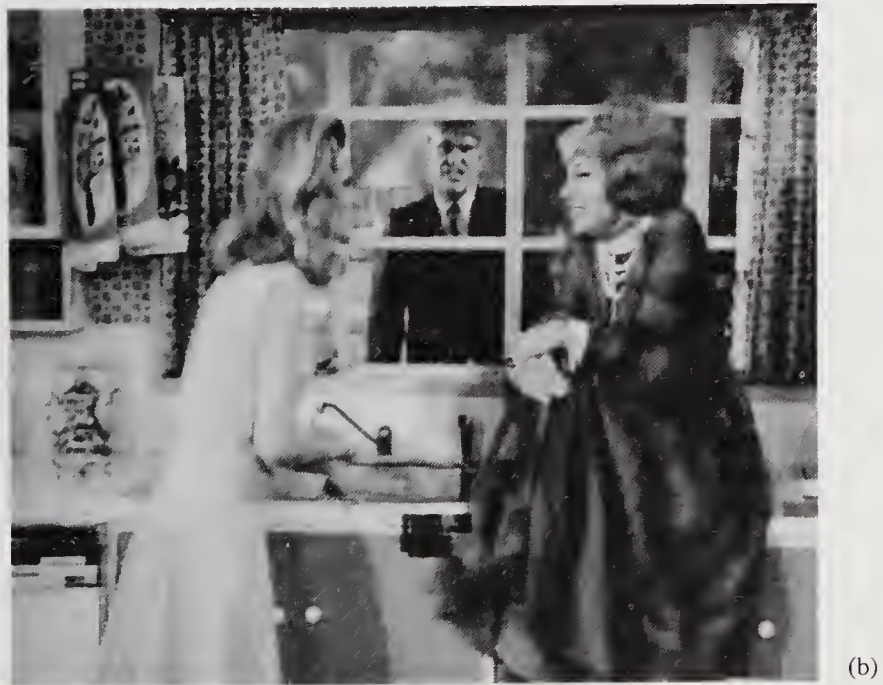
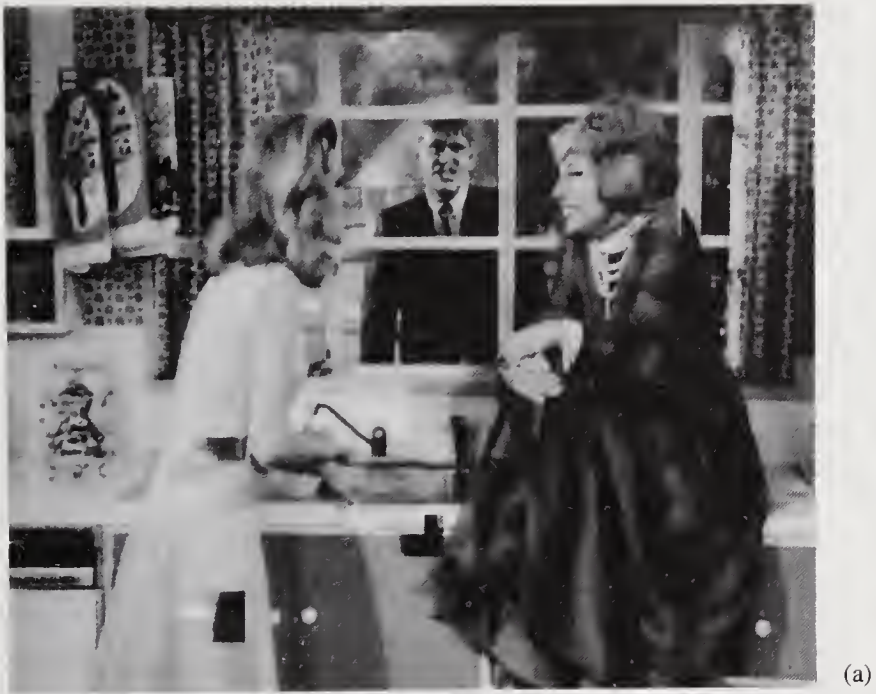


Figure 13 Cell loss at one frame after the lossy scene cut frame , a) unconstrained and b) optimum constrained.



(a)



(b)

Figure 14 Cell loss at five frames after the lossy scene cut frame , a) unconstrained and b) optimum constrained.

since at least 3 macroblocks in a frame are intraframe coded, it may take 132 frames (nearly 5-6 s) till the effect of cell loss can disappear.

The same scene cut frame under the constrained P/M of 2.25 ratio was exposed to 4.5% cell loss. Figure 12b shows the quality of the image at the scene cut, where the cell loss occurred first. It is not surprising that due to smaller cell loss rate, the picture quality is better than that of Figure 12a. In Figure 12b apart from the cell loss artefacts, picture quality in non-lossy area, due to the constraint on the PBR, is poor. However, impairments due to the bit rate constraint (larger quantiser step size) do not appear worse than the cell loss artefacts. At one frame after the scene cut, the quantiser step size is set back to its nominal value. Since the decoder is aware of this change, the picture quality, which was impaired due to the bit rate constraint, improves back to normal. Figure 13b and 14b illustrate the pictures at one frame and five frames respectively after the lossy scene cut frames, where the effect of the bit rate constraint distortion is removed, but that of the cell loss is still present. These pictures in the non-lossy areas exhibit the same quality of the uncontrolled case of Figure 12a. Considering that image sequences are displayed at rates of 25-30 frames per second, the temporary impairments due to the PBR constraint is hardly noticeable, but that of cell loss, similar to the uncontrolled case can last for a long time, till the whole frame is updated, as shown in Figure 15. Since the cell loss in this case is small, the PSNR of the sequence with cell loss is not significantly different from that of without cell loss at scene cuts. However, subjectively the cell loss artefacts are more disturbing than the coding distortions.

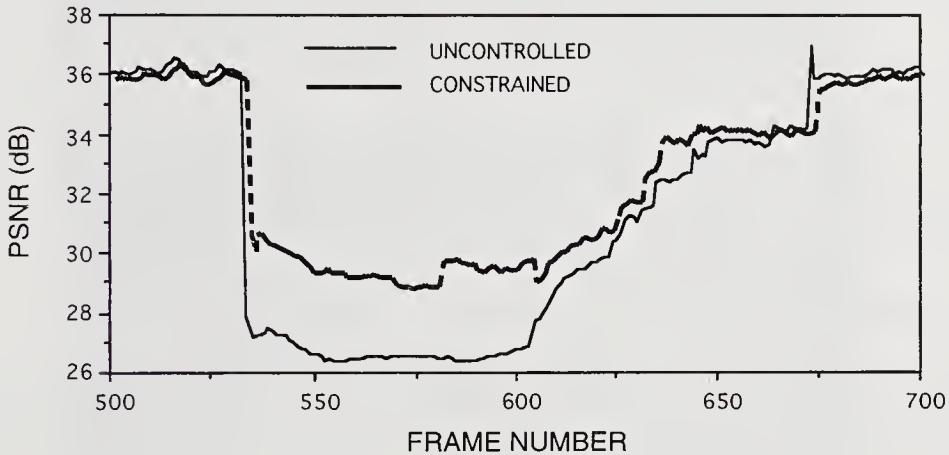


Figure 15 PSNR of the reconstructed video after the occurrence of cell loss at a scene cut.

7.2 Cell Loss in a Non-scene cut Frame

Similar to the scene cut experiment, it was assumed the lost cells are only confined in one interframe coded picture. Figure 16 illustrates the PSNR of the decoded sequence and Figures 17a and 17b show the image quality of a single interframe coded picture at the instant of cell loss for both unconstrained and constrained P/M ratios. Due to smaller cell loss rate under the constrained P/M, the picture degradation is very marginal. The artefacts caused by the cell loss are less disturbing than those of the scene cut frames due to the fact that lost cells do not carry significant information. Furthermore, in the constrained P/M there are no coding impairments in the non-lossy areas since no constraint is imposed on coding of this frame.

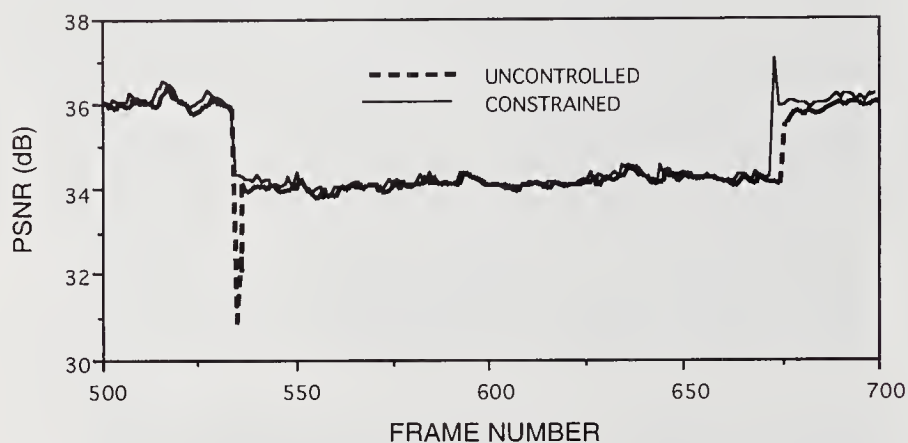


Figure 16: PSNR of the reconstructed video after the occurrence of cell loss at an interframe coded frame.

8 CONCLUSIONS

A method of shaping the video traffic generated by an H.261 type VBR video codec was proposed. The proposed mechanism incorporates a control function to regulate both mean and peak bit rates. At the intraframe coded frames, where the PBR are generated, the video encoder limits its generated bit rate through a leaky bucket mechanism. The bit rates are controlled by adjusting the quantiser step size of the encoder. The adjustment of the quantiser step size is based on the comparison between the actual generated bit rate and the target bit rate within a specified window duration. Decision on the adjustment is made at the start of each video frame. By defining MBR, PBR (P/M ratio), the encoder is able to select a minimum suitable quantiser step size for coding.

It was demonstrated that there is an optimum value of PBR for a given MBR (optimum P/M), where both network and perceived image quality are optimised. This PBR is less than the unconstrained PBR generated by video codecs, and is the value that can be declared by the user.

From the coding point of view, the performance of the shaping mechanism under optimum selection of its parameters, is better than the uncontrolled method both subjectively and in terms of PSNR. It was shown that for a typical video incorporating scene cuts the cell loss rate can be as low as one sixth of the unconstrained methods at low link utilisation.



(a)



(b)

Figure 17 A non-scene cut frame (interframe coded) with cell loss rate of a) 27% unconstrained and b) 4.5% optimum constrained.

9 REFERENCES

- Bae J. and Suda T. (1991) Survey of Traffic Control Schemes and Protocols in ATM Networks. *Proceedings of IEEE*, **14**, 197-204.
- Ghanbari M. and Seferidis V. (1993) Effects of scene cuts on ATM Video. *IEE Electronics Letters*, **30**, 578-579.
- Habib I. W. and Saadawi T. N. (1992) Multimedia Traffic Characteristics in Broadband Networks. *IEEE Communications Magazine*, **30**, 48-54.
- Harasaki H. and Yano M. (1993) A study on VBR coder control under Usage Parameter Control". *Fifth International Workshop on Packet Video*, Berlin.
- Heek H. (1993) A Traffic Control Algorithm for ATM Networks. *IEEE Transactions on Circuits and Systems for Video Technology*, **3**, 182-189.
- Hughes C. J., Ghanbari M., Pearson D. E., Seferidis V. and Xiong J. (1993) Modelling and Subjective Assessment of Cell Discard in ATM Video. *IEEE Transactions in Image Processing*, **3**, 212-222.
- IEEE Communications Magazine* (1991) Special Issue: Congestion Control in High Speed Networks, **29**.
- IEEE Network Magazine* (1992) Special Issue: Congestion Control in High Speed Networks, **6**.
- ITU-T: *Recommendation H.261* (1990) Video coding for audiovisual services at $p \times 64$ Kbit/s.
- ITU-T: *Recommendation I.361* (1991) B-ISDN ATM Layer Specification, Geneva 1991.
- ITU-T: *Recommendation I.371* (1992) Traffic Control and Congestion Control in B-ISDN.
- Kawashima M. and Tominaga H. (1993) A study on VBR video transmission under the Usage Parameter Control. *Fifth International Workshop on Packet Video*, Berlin.
- Niestegge G. (1990) The 'Leaky Bucket' Policing Method in the ATM Networks. *International Journal of Digital and Analog Communications Systems*, **3**, 187-197.
- Ratheb E. P. (1991) Modelling and performance comparison of policing mechanisms for ATM Networks. *IEEE Journal Selected Areas in Communications*, **9**, 343-350.
- Ratheb E. P. (1993) Policing of Realistic VBR Video Traffic in an ATM Network *International Journal of Digital and Analog Communications Systems*, **6**, 213-226, 1993.
- ITU-T SGXV, Working Party XV/4 (1989) Specialists group on coding for visual telephony:, Description of Reference Model 8 (RM8).
- Roberts J. (1992), *COST 224-Performance evaluation and design of multiservice networks*. Commission of the European Communities.
- Sidi, M. Liu, W. Z. Cidon, I. and Gopal I. (1989) Congestion control through input rate regulation. *Proceedings of IEEE GLOBECOM*.

10 BIOGRAPHIES

Anastasios Dagiuklas was born in Greece in 1967. He received the Engineering Degree from the University of Patras, Greece, in 1990, the M.Sc. from the University of Manchester, UK, in 1992 and the Ph.D. from the University of Essex, UK, in 1995, all in Electrical Engineering. During the summer of 1995, he was a visiting researcher to British Telecom Labs, Ipswich, UK, investigating VBR video transmission over ATM networks. His research interests include

congestion control and traffic management in ATM networks, packet video, transmission of multimedia services over high speed networks and Internet.

Dr. Dagiuklas is a member of IEEE.

Mohammad Ghanbari was born in Iran in 1948 and received the BSc in Electrical Engineering from Aryamehr University of Technology, Tehran, Iran in 1970, and the MSc in Telecommunications and Ph.D. in Electronics from University of Essex, UK, in 1976 and 1979 respectively. From 1970-75 and 1979-86, he worked at the Iranian Radio and Broadcasting. In 1986, he joined the image processing research group in the Department of Electronic Systems Engineering, University of Essex, as a Research Fellow, investigating video codecs for ATM networks. He became a Lecturer at the same department in 1988 and was promoted to Senior Lecturer and Reader in 1993 and 1995, respectively. His research interests are: video bandwidth compression, motion estimation, video coding for ATM networks and multimedia communications. He has pioneered the two-layer video coding for ATM networks.

Dr. Ghanbari is a Chartered Engineer and a member of IEE and IEEE.

Bee June Tye received the Diploma in Electronics and Communications Engineering in 1988, the Advance Diploma in Computer and Communications Systems in 1993 all from the University of Singapore and the Msc. in Telecommunication and Information Systems in 1994, from the University of Essex, UK. She is currently with the AT&T in Singapore responsible for software development for the cordless telephony.

PART EIGHT

Performance Modelling Studies

STUDY OF THE PERFORMANCE OF AN ATM CLOS SWITCHING NETWORK BASED ON THE COMPOSITE TECHNIQUE

*Fiche G., Le Palud Cl., Rouillard S.
Alcatel CIT, 4 rue de Broglie 22304 LANNION France
tel. 3396047316 fax 3396048300*

Abstract

In this article we study the performance of a real ATM switching network designed for incorporation into existing switching systems. Because of its architecture based on the ATM Composite technique, this new network will give access to Broad-Band services (images, data, etc.) while still remaining compatible with the constraints inherent in Narrow-Band services such as speech or high-quality sound.

The focus here is on the investigation of the system behaviour to determine dimensioning and call acceptance rules yielding a high-performance network *for both Narrow-Band and Broad-Band services*.

To make a complete study of the performance of such a network, three calculations are carried out in turn:

- Blocking of 64Kb/s connections, i.e. the probability that a 64Kb/s call will be rejected because no route is available (shortage of composite ATM cells);
- Blocking of Broad-Band services connections, i.e. the probability that a VBR or CBR-type call will be rejected because no bandwidth is available;
- The Cell Delay Variation (CDV) of the cells carrying the services.

The main contributions of this paper are first, the application of overflow traffic theory, which enables us to give an exact solution of the number of cells required for handling 64 Kbit/s services with the ATM Composite technique, and second, the determination of a very accurate formula for the calculation of the blocking probability for Broad-Band services which yields quite a good network dimensioning rule and efficient call acceptance algorithms.

Keywords

ATM, Broad-Band, blocking, call acceptance, cell delay variation, CLOS network, composite technique, Narrow-Band, performance, switching network, statistical multiplexing.

INTRODUCTION

Evaluating the performance and determining call acceptance procedures for ATM switching networks is critical for the choice of an architecture. This study evaluates an ATM switching network using the composite technique for narrowband services associated with statistical multiplexing for Broad-Band services.

The architecture of the network being studied is shown in the diagram below.

In this network, 64Kb/s services (Narrow-Band services) are processed using the ATM Composite technique (ref.1). The principle of this technique is to combine time slots of incoming PCMs from the same ATM Composite matrix (T/AC), intended for the same outgoing matrix (AC/T), on one or more cells reserved within an established virtual circuit for that direction.

On the other hand, Broad-Band services - described by a three-state model (Passive /On/Off) - are multiplexed statistically on entrance to the network (MUX), obeying a rule for calls acceptance which guarantees the quality of the service at the cell and call levels. In the same way as for 64Kb/s services, virtual circuits are established per call within the core of the network.

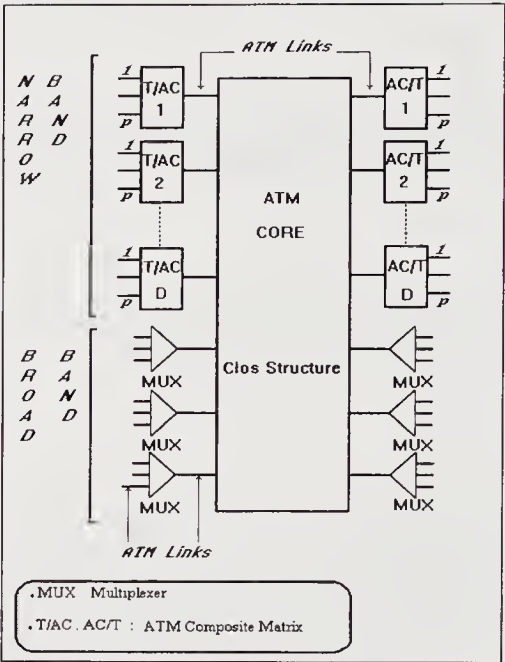


Figure 1

Basically the core of the ATM network has a CLOS (ref.2) structure, which ensures that for any service accessing an incoming link it will be possible to establish a path ,i.e. a virtual circuit within the network. At this level, the only constraint is the network crossing delay for the cells depending on the load of the links.

The performance of such a network is therefore described by the blocking probability for 64Kb/s and Broad-Band calls, and by the crossing delay of the cells. From the blocking of calls we can deduce the permissible load on the network's internal links. This load is then used to determine the crossing delay.

1. Study of blocking of 64Kb/s connections

1.1 The Narrow-Band switching matrix and the ATM Composite

Using the ATM technique to transport 64Kb/s services carried on PCM frames requires basically a method of adapting information from frame format to ATM format. To solve this problem, there is an advantage in using an adaptation layer ("composite") in which the payload of an ATM cell is made up of time slots from several 64Kb/s channels.

This technique involves creating, on demand, virtual circuits (cells VC) between the input and output matrices (T/AC and AC/T) connecting the PCMs to the ATM network. The time-slots of PCMs from the same matrix to the same output matrix are gathered in one or more cells carried by virtual paths (VP) ; each connection being fully defined by its unique VPI (Virtual Path Identifier) and VCI (Virtual Channel Identifier).

The principle is as follows:

1) In the incoming T/AC matrix (I), a connection is set up between:

- the time slot of a 64Kb/s connection carried on an incoming PCM frame (two bytes per time slot)
- and two free bytes contained in the information field (the payload) of an ATM cell responsible for transporting information from that incoming T/AC matrix (I) to the appropriate outgoing AC/T matrix (O).

2) In the ATM switching network a Virtual Path (VP) connection transports the various cells between matrices (I) and (O).

3) In the outgoing AC/T matrix (O), a connection is established between:

- the two bytes contained in the payload of an ATM cell received by AC/T matrix (O)
- and the time slot of the outgoing 64Kb/s connection carried on a PCM frame sent out by AC/T matrix (O).

These unitary connections give rise to a search for free space in one or more set up between matrices (I) and (O). Twenty three (23) spaces are available per VC, because among the 48 bytes of payload, two bytes are reserved for AAL1 functions. If there is not enough space, a new VC can be set up. However the number of VCs in use is limited by the capacity of the ATM link; for example on a 622 Mbit/s ATM link, only 183 cells of 53 bytes are available every 125 μ s.

It could then happen that no space would be available in the cells already open for a given destination and that the opening of a new cell would be impossible in spite of free places for other directions. In this case, the system will not be able to accept an incoming call which will then be blocked.

1.2 Modeling

Let us consider one of the D number of T/AC input matrices. From the combined p PCMs which it is connected to, it will receive traffic at intensity A generated by subscribers and circuits access units. As only a negligible amount of traffic is rejected by these units, which concentrate the traffic of a large number of sources, the traffic offered to this matrix can be taken to follow a Poisson distribution.

This matrix then offers its traffic to the D AC/T matrices at the output stage. It distributes them with equal probability, and so the traffic offered in any given direction will follow a Poisson distribution of intensity A/D.

Now, let us take the case of a network with D directions being offered a traffic A such as, in average, one cell with 23 connections in any direction will suffice to carry the traffic A/D. Fluctuations in the traffic offered may then imply that supplementary cells could be needed in some of the D output directions. For a given direction, the effect is as if we had the following system of service:

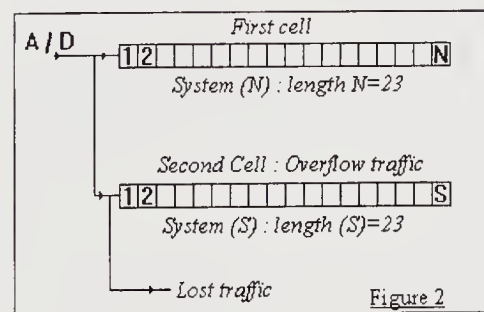


Figure 2

Traffic of intensity A/D is initially offered to the first cell in one of the D directions. Any call arriving when all 23 connections in this cell are occupied will be offered to a second cell. We then have a traffic overflow, excess traffic from the first cell being offered to the second. This procedure could in turn lead to traffic being rejected by the second cell and lost completely, but in fact in our study the total loss probability is so small as to be taken as zero.

1.2.1 Analysis of space for transition probabilities:

The system described above corresponds exactly to an overflow system consisting of two sub-systems which we shall call (N) and (S). A state (n,s) can be described in terms of the number of connections n occupied in (N) where $n \leq N$, and s connections occupied in (S) where $s \leq S$. The behaviour of incoming connections (calls) depends on the state of (N) and (S). The calls can be said to be directed to (N) in the first instance, and when (N) is full they are redirected to (S). This means that the state of (N) is independent of (S), but not vice versa.

This type of system has been studied by many authors. In the next sections we will follow Brockmeyer's analysis (ref.3).

The behaviour of (N) is a birth and death process within a number of states limited to N ; that of (S) is a pure death process so long as $n < N$, because (S) is not then fed with calls - in fact, so long as $n < N$, (S) only finds its calls coming to an end. The only time that (S) receives calls is when (N) has all its 23 connections engaged. The process then becomes a birth and death one.

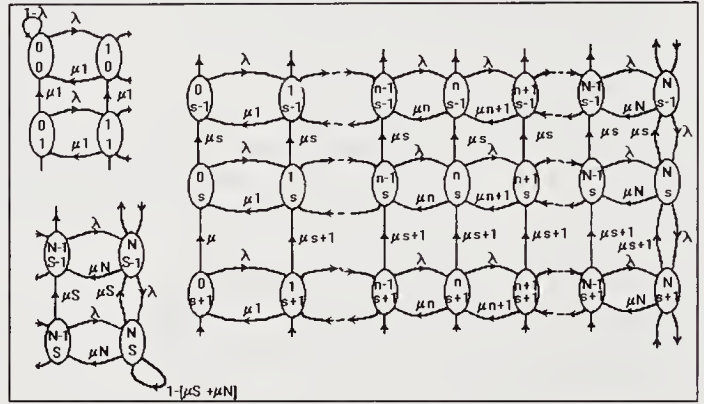


Figure 3

The state graph below (Figure 3) describes the space of probabilities. From it we can derive the "equations of future" and the system's state equations.

We note P_s^n the probability of state (n,s) .

The "equations of future" derived from the graph are:

$$\begin{aligned}
 P_0^0(t + \Delta t) &= P_1^0(t) \cdot \mu_1 \Delta t + P_0^1(t) \mu_1 \Delta t + P_0^0(t) (1 - \lambda \Delta t) \\
 P_s^n(t + \Delta t) &= P_s^{n-1}(t) \cdot \lambda \Delta t + P_{s+1}^n(t) \mu \Delta t + P_s^{n+1}(t) \mu_{n+1} \Delta t + P_s^n(t) [1 - (\lambda + \mu n + \mu s) \Delta t] \\
 P_s^N(t + \Delta t) &= P_s^{N-1}(t) \cdot \lambda \Delta t + P_{s-1}^N(t) \lambda \Delta t + P_{s+1}^N(t) \mu_{s+1} \Delta t + P_s^N(t) [1 - (\lambda + \mu N + \mu s) \Delta t] \\
 P_s^N(t + \Delta t) &= P_{s-1}^N(t) \cdot \lambda \Delta t + P_s^{N-1}(t) \lambda \Delta t + P_s^N(t) [1 - (\mu N + \mu s) \Delta t]
 \end{aligned} \tag{1}$$

and the state equations corresponding to statistical balance are then:

$$\begin{aligned}
 P_0^0 \left(\frac{A}{D} \right) - P_1^0 - P_0^1 &= 0 \\
 P_s^n \left(\frac{A}{D} + n + s \right) - P_s^{n-1} \frac{A}{D} - P_s^{n+1} (n+1) - P_{s+1}^n (s+1) &= 0 \\
 P_s^N \left(\frac{A}{D} + N + s \right) - P_s^{N-1} \frac{A}{D} - P_{s-1}^N \frac{A}{D} - P_{s+1}^N (s+1) &= 0 \\
 P_s^N (N + S) - P_s^{N-1} \frac{A}{D} - P_{s-1}^N \frac{A}{D} &= 0
 \end{aligned} \tag{2}$$

1.2.2 Solution of the set of equations

To solve the above set of equations, Brockmeyer introduces the polynomial S_r^m defined by:

$$S_r^m(A/D) = \sum_{v=0}^m \frac{A/D^{m-v}}{(m-v)!} \binom{r-1+v}{v} \quad \text{and} \quad S_r^m = 0 \quad \text{if } m < 0 \text{ or } r < 0 \quad (3)$$

The solution is then written thus:

$$P_i^j = \sum_{x=0}^{S-1} (-1)^x K_{i+x} \binom{i+x}{i} S_{i+x}^{j-x}$$

Where :

$$K_k = \sum_{r=k}^S (-1)^{r-k} \binom{r-1}{k-1} a_r \quad \text{and} \quad K_0 = \frac{1}{S_1^{N+S}} \quad (4)$$

and :

$$a_r = \frac{1}{S_1^{N+S} S_r^N} \sum_{v=r}^S \binom{v-1}{r-1} S_0^{N+v}$$

The distribution of overflow given by : $Q(i) = \sum_{j=0}^N P_i^j$

becomes :

$$Q(i) = \sum_{x=0}^{S-i} (-1)^x K_{i+x} \binom{i+x}{i} S_{i+1+x}^{N-x} \quad (5)$$

And so, in our application, the probability P that more than one cell will be used is:

$$P = \sum_{i=1}^{23} Q(i)$$

$$P = \sum_{i=1}^{23} \left[\sum_{x=0}^{S-i} (-1)^x K_{i+x} \binom{i+x}{i} S_{i+1+x}^{N-x} \right] \quad (6)$$

where $S=N=23$

The total number of cells required in all D directions can then be easily obtained. Since traffic is offered independently to each of the D directions, the distribution of the number of cells required is given by the binomial law :

$$P(N_c) = C_D^k P^k (1-P)^{(D-k)}$$

where $P(N_c)$ is the probability that exactly $N_c=2k+(D-k)$ cells will be engaged and P , the probability given by (6). It will be said that there is call blocking whenever N_c is greater than a given value N_{\max} ($N_{\max} = 183$ in our study which is the maximum number of cells available every 125 μ s on a 622 Mbit/s ATM link).

The average number of cells engaged is : $\overline{N_c} = [(1 - P) + P * 2]D$

The occupancy rate (ρ) of the ATM links will be derived from this number ; i.e. : the probability of a cell to be engaged is :

$$\rho = \frac{\overline{N_c}}{N_{\max}}$$

GENERALISATION

The result above can be applied to cases where x cells are systematically used in each direction ($x = \lfloor A/23D \rfloor$); the value of N is then set at $23.x$ and the value of S remains at 23. In this case it is assumed that the probability of using fewer than x cells or more than $x+1$ cells in each direction is negligible.

The distribution of the total number of cells required in all D directions is then :

$$P(N_c) = C_D^k P^k (1 - P)^{(D-k)} \quad (7)$$

where $P(N_c)$ is the probability of having exactly $N_c=k(x+1)+(D-k)x$ cells engaged, and the average number of cells engaged is :

$$\overline{N_c} = [(1 - P)x + P(x + 1)]D \quad (8)$$

1.3 Application

Being given a loss probability, a study of the performance of a switching network would consist in the determination of the traffic load to be offered and in the calculation of the mean number of used cells (which will be involved in CDV calculation).

We will therefore apply the results derived above and compare the numerical values obtained against those found by simulation.

The following two graphs give an example of the results we achieved. As can be seen, the agreement between the simulation and the calculation is perfect.

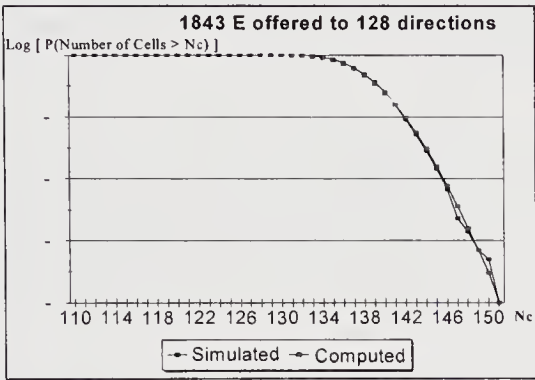


figure 4

The first graph, derived from result (7), enables us to deduce from the capacity of the multiplex used, the probability for calls to be blocked. With a multiplex of 622 Mb/s (183 cells), this probability can be seen to be negligible, even with loads of 1843 E per TCA on incoming PCMs, as in our example of implementation.

This is the type of result which will be used to dimension the system (to determine the number of PCMs that can be connected).

The second graph (derived from result (8)), with its unusual shape, gives the cell load of the links inside the network for a given load of PCMs and a given number of TCA matrices in use. The particular shape actually arises as follows : as the number of directions (matrices) grows, the number of cells strictly required and the number of additional (overflow) cells tend to increase. However, for certain configurations, the cells are more or less filled to their optimum potential, allowing the same number of cells to be used for different numbers of directions (matrices).

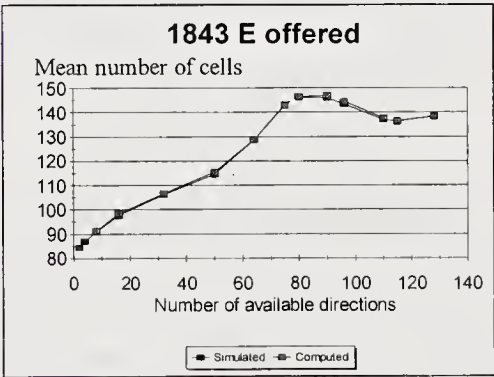


Figure (5)

From these two graphs, we can see first that in the case of 1843 E offered to 128 directions, the probability of needing more than 150 cells is 10^{-4} (Figure 4) and, as a consequence, the probability for needing more than 183 cells (call blocking) is negligible, and second that in this case, the mean number of used cells is 138 (Figure 5).

We deduce from this last result that the occupancy rate of a cell is equal to $\rho = 138/183 = 0.754$.

2. ATM sources multiplexing / Blocking for the Broad-Band service connections

In this second part we shall consider the saturation probability for the 622 Mb/s links by ATM sources (terminals) with variable rates (VBR). Our research looks at a system consisting of K different types of sources, numbered N_1, N_2, \dots, N_k , with variable rates. The operating principle we have adopted can be summarised as follows:

Sources connected to the system generate calls which will be accepted or not, depending on their traffic characteristics and how busy the multiplex is.

Indeed, we suppose the system to be able to identify at any moment the number of calls of each type already accepted in the network, on each link. This enables us to know at any moment the statistical characteristics of the traffic offered to the multiplexes. A call of type j will then be accepted if there is sufficient bandwidth left to take it, otherwise it will be rejected.

A counter is incremented for each type of call whenever a new call is accepted, and decremented when the call ends. A table describes the combinations n_1, n_2, n_k of active sources of type t_1, t_2, t_k , which are compatible with a given maximum probability of saturation of the multiplex, as defined below. A new call is then only accepted if its characteristics are compatible with the content of the table.

In a first step, for each type of sources, we calculate the probability that multiplex will be saturated given that, n_1 of the N_1 sources of type 1 connected, n_2 of N_2 sources of type 2, ... n_k of N_k sources of type K, are active. Next, we establish a dimensioning rule which allows us to determine easily the number of sources of each type which can be connected to the multiplexer (N_1, N_2, \dots, N_k).

2.1 Multiplex saturation probability

2.1.1 Definition of sources

Here we shall consider a model of sources with three states (figure 6) suggested by the ATM Forum Technical Committee. A source can be either ACTIVE (making a call), or PASSIVE.

When it is active, the source generates a series of packets or bursts (ON) separated by short pauses (OFF).

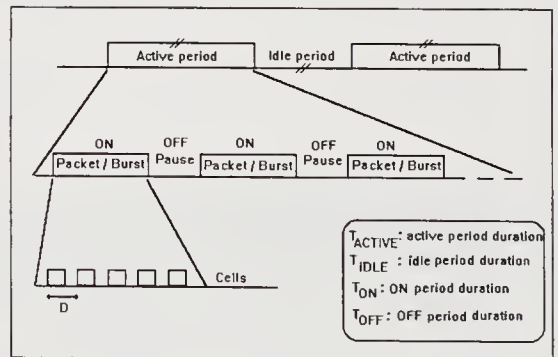


Figure 6

Now let us consider sources of different types, along the lines of the three-state model just described but with transmission rates f_i which differ from each other. In particular we might consider the connection of Distributed Computing Environment (DCE) sources at 2 Mb/s, 4 Mb/s, 30 Mb/s or even 150 Mb/s. These different frequencies will bring about variations in the values of T_{ON} and T_{ACTIVE} . We should note that this model applies to SBR-type sources and equally to CBR/DBR-type sources, though there the ON state is the same as the ACTIVE state.

Since the multiplex is characterised by a rate f , each source as seen by the multiplex is characterised by $D_i = f/f_i$ time-slots spaced out over the cells that constitute a burst (i.e. the multiplex can handle D_i sources of type i at the same time).

We will assume that $D_{\max} = \max_{1 \leq i \leq k} \{D_i\}$ and $d_i = D_{\max}/D_i$.

Thus the multiplex has to handle traffic from N_1 sources of type 1, generating calls that occupy d_1 units of the bandwidth, N_2 sources of type 2, generating calls that take up d_2 units,... N_K sources generating calls that take up d_k units; the total number of bandwidth units available on the multiplex being given by D_{\max} .

2.1.2 Blocking probability calculation

Having defined the model of sources, we can now calculate the probability of having r_1 sources of type 1, r_2 sources of type 2,, r_k sources of type k , all in the ON state when we know that n_1, n_2, \dots, n_k sources of each type are active. When only one type of source is used, this probability is given by Engset's Law (we are only considering here congestion in time):

$$P_{ON(n)} = \frac{C_M^n p_{ON}^n (1-p_{ON})^{(M-n)}}{\sum_{i=0}^N C_M^i p_{ON}^i (1-p_{ON})^{(M-i)}} \quad P_{ON(n)} \text{ is the probability of having } n \text{ } (n \leq N) \text{ sources ON, knowing that } M \text{ } (n \leq M) \text{ are active and that } N \text{ } (N \leq M) \text{ sources at most can be ON at the same time.} \quad (9)$$

In the event of k different types of sources being used, the probability of having r_1, r_2, \dots, r_k sources ON, when we know that n_1, n_2, \dots, n_k sources of each type are active, can be expressed as previously as the ratio between the probability of favourable cases and the probability of all possible cases.

Thus we obtain:

$$P_{ON(n_1, n_2, \dots, n_k)}(r_1, r_2, \dots, r_k) = \frac{\prod_{i=1}^{i=k} C_{n_i}^{r_i} p_{ONi}^{r_i} (1-p_{ONi})^{(n_i-r_i)}}{\sum_{r_1}^{l_1} \sum_{r_2}^{l_2} \dots \sum_{r_k}^{l_k} \left[\prod_{i=1}^k C_{n_i}^{r_i} p_{ONi}^{r_i} (1-p_{ONi})^{(n_i-r_i)} \right]} \quad (10)$$

where : $l_i = \left\lceil \frac{D_{\max} - \sum_{j < i} d_j r_j}{d_i} \right\rceil$

The result above is in fact simply the expression of Engset's generalised law, very similar to Erlang's generalised law given in (ref.4), which can be easily obtained by making the values of n_i ($1 \leq i \leq k$) tend to infinity.

The probability of having r units of bandwidth engaged, knowing that n_1, n_2, \dots, n_k sources are active, is then expressed as :

$$P_r = \sum_{\{(r_1, r_2, \dots, r_k) / r_1 d_1 + r_2 d_2 + \dots + r_k d_k = r\}} P_{ON(n_1, n_2, \dots, n_k)}(r_1, r_2, \dots, r_k) \quad (11)$$

A type of active source j is said blocked when the remaining bandwidth is inadequate, which is the equivalent of having sources r_1, r_2, \dots, r_k all ON, such that :

$$\sum_{i=1}^K r_i d_i > D_{\max} - d_j \Leftrightarrow \sum_{i=1}^K r_i f_i > f - f_j \quad (12)$$

Therefore if P_{Bj} is the probability that sources of type j are blocked, P_{Bj} is then expressed as:

$$P_{Bj} = \sum_{r=D_{\max}-d_j+1}^{D_{\max}} P_r$$

$$P_{Bj} = \frac{\sum_{r=D_{\max}-d_j+1}^{D_{\max}} \left\{ \sum_{\{(x_1, x_2, \dots, x_k) / r_1 d_1 + r_2 d_2 + \dots + r_k d_k = r\}} \prod_{i=1}^{i=k} C_{n_i}^{r_i} P_{ON_i}^{r_i} (1 - P_{ON_i})^{(n_i - r_i)} \right\}}{\sum_{\{(x_1, x_2, \dots, x_k) / r_1 d_1 + r_2 d_2 + \dots + r_k d_k \leq D_{\max}\}} \left[\prod_{i=1}^{i=k} C_{n_i}^{r_i} P_{ON_i}^{r_i} (1 - P_{ON_i})^{(n_i - r_i)} \right]} \quad (13)$$

Given a maximum call blocking probability value (P_{bloc_i}), for each type of source i , the formula (13) above may be used to determine all the combinations (n_1, n_2, \dots, n_k) of active sources which satisfy the relationship $P_{Bi} \leq P_{bloc_i}$. The intersection of the sets thus defined for each type of source makes up a set \mathcal{E} of combinations (n_1, n_2, \dots, n_k) of active sources, such that : $\forall i, P_{Bi} \leq P_{bloc_i}$.

☞ The set \mathcal{E} and the count of the number of active sources will form the basis of the call acceptance procedure (sources in active state).

☞ As may be seen thereafter, it is easy to verify, particularly in the case of identical ON-OFF sources, that our result (13) gives a very good evaluation of the "knee" of the distribution of cells in a multiplexing waiting queue.

2.1.3 Practical meaning of the formula

Let us consider the superposition of ON-OFF bursty sources.

It is widely recognised that, in a multiplexer with an infinite queue, the probability distribution of waiting cells, $P(>x)$, can be divided in two parts : one corresponding to the cell component and the other to the burst component.

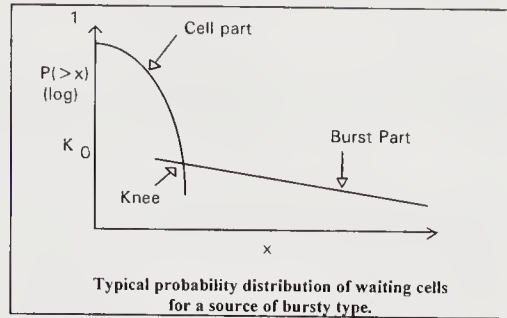


figure 7

As demonstrated in (ref. 5 - P.442), the cell part can be calculated very precisely by applying the $(\sum N_i * D_i / D / 1)$ formula to the mean bit rate and also, in the case of low load, a good approximation is obtained by using the M/D/1 formula.

The burst part has also been investigated a lot in literature (Ref. 5, 6, 7, 8, 9). For this part of the curve, the results derived in many studies show that the queue length probability distribution essentially depends on bursts length. In the case of short bursts the slope of the curve will be rather high and therefore, increasing the buffer capacity will enable us to limit loss probability.

However, it is important to notice that if bursts are long enough (which is the more general case), the slope will be so low that extremely large queues would be necessary. In this case, increasing buffer size will not provide significant multiplexing gain, and moreover this is not compatible with services with real time constraints. (It is preferable to have queues with different priorities) .

It seems therefore realistic to use formula (13) to determine the traffic load offered to the multiplexer in order to remain just before the "knee" of the curve (Point K_0), in the cell part.

Indeed, as explained in (ref.5 - P.448), we are, in this part, in a situation where the probability of an overflow of the capacity of the multiplexer rate is negligible whereas, in the burst region, the behaviour of the queue is governed by the fact that a total saturation of the multiplex occurs during a peak period : the probability for having too many bursts simultaneously active is high compared to the probability of congestion by cells.

That is the reason why the formula (13) derived to calculate the probability for sources to be blocked gives a very good evaluation of the "knee" of the distribution of cells in a multiplexing queue (Point K_0 - figure 7). The accuracy of this assertion has been verified through several comparisons, in particular for homogeneous ON-OFF traffic sources as demonstrated hereafter. We shall now call it: burst level call blocking probability.

Table 8 sums up the results of our comparisons against simulation examples found in literature or achieved internally.

n , Dmax, Pon	Simulation Value ¹	Calculated Call Congestion ₂	Calculated Time Congestion ₃
[1] 80, 48, 0.35	2 10 ⁻⁶	1.81 10 ⁻⁶	2.94 10 ⁻⁶
[2] 33, 16, 0.375	5 10 ⁻²	5.25 10 ⁻²	6.50 10 ⁻²
44, 16, 0.285	6.5 10 ⁻²	6.35 10 ⁻²	7.26 10 ⁻²
75, 16, 0.166	7 10 ⁻²	6.75 10 ⁻²	7.23 10 ⁻²
[3] 36, 12, 0.1	1 10 ⁻⁴	7.39 10 ⁻⁵	9.98 10 ⁻⁵
36, 12, 0.2	3 10 ⁻²	2.04 10 ⁻²	2.46 10 ⁻²
[4] 100,15,0.0435	2 10 ⁻⁵	1.93 10 ⁻⁵	2.17 10 ⁻⁵

Table 8

¹ ■ Due to the fact that we only found curves of simulations, values given in this column are approximate ones.

² ■ Call congestion is obtained in substituting n_i-1 to n_i in formula (13)

³ ■ Time congestion has been derived from formula (13)

[1] Information technologies and sciences. *COST 224*, p.185, 1992.

[2] ANICK D., MITRA D. SONDHI M .M. Stochastic Theory of a Data-Handling System with Multiple Sources. *The BELL Technical Journal*, Vol.61, N°8, pp 1871-1894, Dec. 1981.

[3] Internal Simulations

[4] YANG T., TSANG D. H. K. A Novel Approach to Estimating the Cell Loss Probability in an ATM Multiplexer Loaded with Homogeneous ON-OFF Sources. *IEEE Transactions on COMMUNICATIONS*, Vol.43, N°1, pp 117-126, Jan. 1995.

As may be seen in table 8, in the case of homogeneous sources, the agreement between our calculations and the simulation examples we found is excellent.

Moreover, we have noticed the same agreement in the case of heterogeneous sources. As may be seen below, we have achieved several calculations for a mix of two classes at a time (50% Type 1, 50% Type 2). Figure 9 deals with the comparison between the probability of total time congestion (calculated from the probabilities of congestion obtained for each type of sources - cf table below) and simulation results found in literature. It is interesting to note that the accuracy of our formula remains excellent even in case of low load.

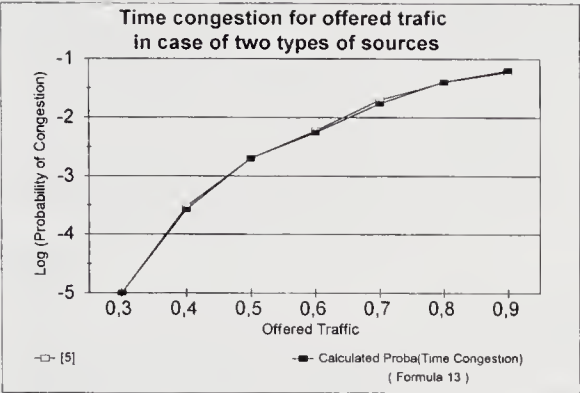


Figure 9

Offered Traffic A	[5]	PB ₁ Formula(13)
0,3	1E-05	9,4E-06
0,4	0,0003	0,000235
0,5	0,002	0,00175
0,6	0,006	0,00468
0,7	0,02	0,0144
0,8	0,04	0,0334
0,9	0,06	0,0519

[5] BAIOCCHI A., BLEFARI-MELAZZI N., ROVERI A., SALVATORE F. Stochastic Fluid Analysis of an ATM Multiplexer Loaded With Heterogeneous ON-OFF Sources : an Effective Computational Approach. *INFOCOM 92*, pp 405-414, 1992.

[5] → Fluid Flow Model Two types of sources.							
Type 1 : P _{ON1} = 0.1, D ₁ = 7							
Type 2 : P _{ON2} = 0.5, D ₂ = 23							
D _{MAX} = 23, d ₁ = 3, d ₂ = 1 A = offered traffic							
A	0,3	0,4	0,5	0,6	0,7	0,8	0,9
n1	8	11	14	16	19	22	24
n2	8	11	14	16	19	22	24

As a consequence of the way of dimensioning we recommended (at point K₀), the traffic accepted on the network links can be modelled as a geometric one. Thus, simple formulae for the dimensioning of the queues within the network matrices may be used and, most important, small buffers are sufficient.

2.2 Call acceptance and System dimensioning

♦ Call acceptance is based on the use of formula (13), which allows to determine the set \mathcal{E} of acceptable combinations (n₁ , n₂ , ..., n_K) of active sources such that the probability of saturation of the multiplex is below a predetermined limit.

The expansion within the Clos network ensures to be able to establish a path on the basis of the peak rate, as far as the sum of the peak rates offered to an entry matrix is less than a maximum value. This value and the expansion required may be determined by formulas such as presented in (ref.10). Under this constraint the core of the network is strictly non blocking. If we accept a negligible call rejecting probability in the ATM core it is furthermore possible to reduce the expansion, and even to increase the multiplexing gain as follows: a path is

established within the network by testing on each link that the new call is compatible with the set of acceptable combinations assuring the maximum saturation probability allowed. In both cases the control leads to remain below the "knee" thus allowing short buffers and simple calculations for the size of the queue.

- ♦ Dimensioning the system, i.e. determining the number of sources of each type that can be connected to the multiplex, is based on the blocking probability allowed for each type of sources.

Having calculated the set \mathcal{E} of acceptable combinations (n_1, n_2, \dots, n_K) of active sources, we can now determine the set of combinations (N_1, N_2, \dots, N_K) of sources connected, such that the probability of obtaining an unacceptable combination (not forming part of \mathcal{E}) (n_1, n_2, \dots, n_K) active sources falls below a pre-determined limit. Once again, this probability is given by Engset's Law because the number of sources which can be active at the same time is limited. Thus, if we give the name \hat{E} to the set of combinations which form the boundary of the set \mathcal{E} of permissible combinations, we obtain :

$$\text{Proba. of refusing a call} = \sum_{(n_1, n_2, \dots, n_K) \in \hat{E}} \frac{\prod_{i=1}^k C_{N_i}^{n_i} P_{ACT_i}^{n_i} (1 - P_{ACT_i})^{(N_i - n_i)}}{\sum_{\{(a_1, a_2, \dots, a_k) / a_1 \leq n_1, a_2 \leq n_2, \dots, a_k \leq n_k\}} \prod_{i=1}^k C_{N_i}^{a_i} P_{ACT_i}^{a_i} (1 - P_{ACT_i})^{(N_i - a_i)}} \quad (14)$$

The system can then be dimensioned according to the following algorithm:

- ① For each type of sources i , the active sources configurations (n_1, n_2, \dots, n_K) are determined such that the burst level call blocking probability P_{Bi} is less than the predetermined value P_{bloc_i} , 10^{-7} for example. (Use of formula (13)).
- ② From the sets of combinations derived for each source, the set \mathcal{E} of acceptable combinations is determined, together with its upper limit \hat{E} .
- ③ Knowing \hat{E} , the set of combinations (N_1, N_2, \dots, N_K) of connectable sources is determined, such that the probability of calls being refused, calculated from the formula (14), falls below a predetermined level (10^{-3} for example).

2.3 Implementation

If on the one hand calculations seem to be complex, on the other hand the implementation of calls acceptance procedures is really simple.

- ♦ A call counter is incremented as each call arrives, and decremented when it ends. The state of the counter (n_1, n_2, \dots, n_K) is then compared with the contents of the set \mathcal{E} of permissible combinations, and sources leading to a combination (n_1, n_2, \dots, n_K) not belonging to \mathcal{E} are rejected (comparison with an engineering table defining the boundary \hat{E} of \mathcal{E} (cf. below)). The use of those predetermined tables seems to be an efficient solution taking into account the relative complexity of the formulas.

Example

Let us take the system with two types of source as in this diagram:

Each type of source is characterised by its rate, and follows the three-state model, described above.

We apply the algorithm previously defined:

Firstly, we calculate the burst level call blocking probability (formula (13)) for each type of source, taking (n_1, n_2) sources to be active. We then draw the graph opposite (Figure 11), showing limiting curves for the permissible configurations compatible with the burst blocking probabilities, for each type of source.

As expected, we find that it is type-2 sources which impose the severest constraints (so this result suggests that we should accept a higher blocking probability for sources with a high rate (type 2)).

The set \hat{E} , including the new call, is then given by the table of limit combinations (n_1, n_2) of active sources such that the burst level call blocking probability for the two types of sources falls below 10^{-7} :

n_1	0	1	2	3	4	5	6	7	8	9
n_2	4	3	3	3	3	3	2	2	2	2

n_1	10	11	12	13	14	15	16	17	18	19	20
n_2	2	1	1	1	1	1	0	0	0	0	0

Having determined this set, we then consider the dimensioning of the system. We therefore apply the formula previously established (formula (14)) to various combinations (N_1, N_2) of connected sources, and note the maximum combinations which give a calculated call rejecting probability lower than 10^{-3} (probability of obtaining the set \hat{E} - Figure 12).

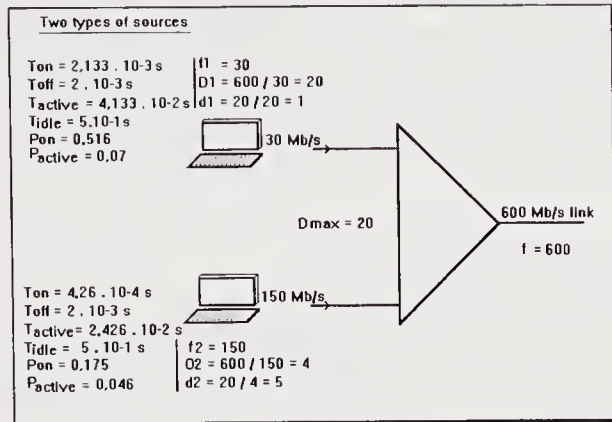


Figure 10

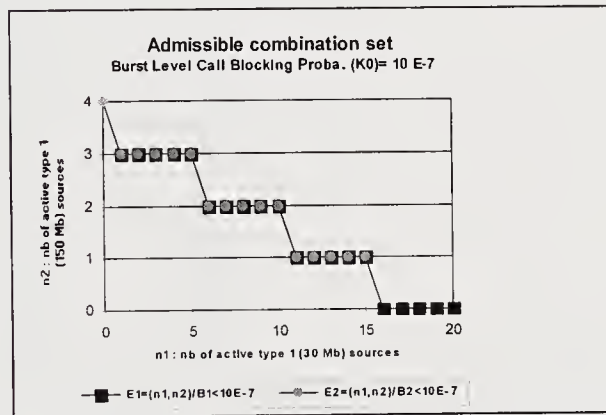


Figure 11

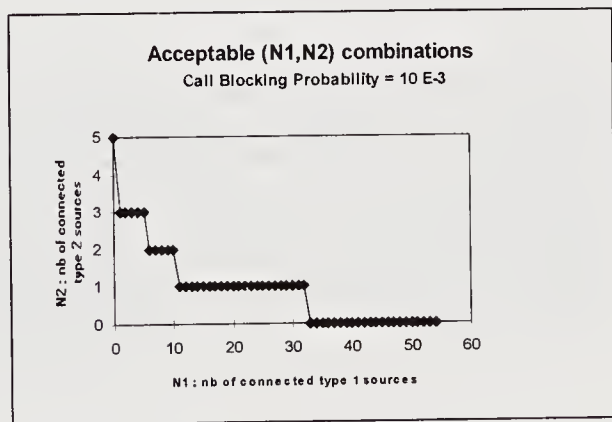


Figure 12

We can therefore conclude, for example, that with a configuration of five type-1 sources and two type-2 sources, the probability that a given source is prevented from transmitting because probability of overload at the multiplex is greater than 10^{-7} , is less than 10^{-3} .

It is furthermore interesting to notice that, for low blocking probability, there is a significant multiplexing gain only if the ratio of the multiplex rate over the source rate is large enough. Otherwise, as it is the case in our example, it is nearly equivalent to dimension on the basis of the peak rate.

3. Evaluation of the CDV

In this last section, we shall calculate the time taken for cells to cross the network, giving us an upper limiting value for the Cell Delay Variation (CDV). The ATM core of the network is a three-stage Clos structure, with expansion. The diagram below shows the configuration adopted.

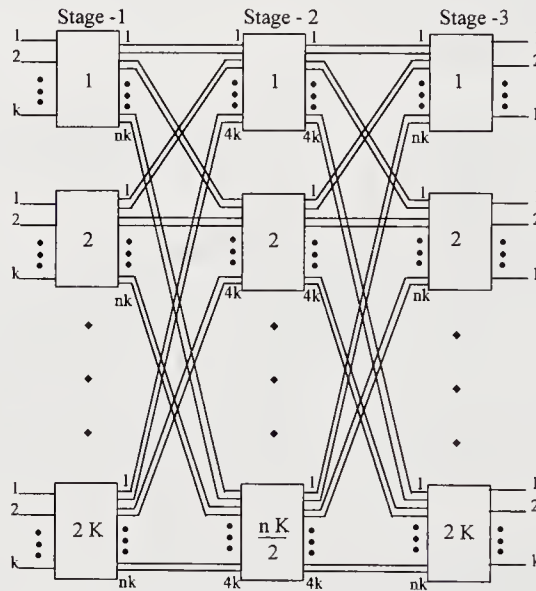


Figure 13

At each stage, there may be a delay as cells from different directions may wait for access to an outgoing direction. Taking the usual hypothesis of independence between stages, the distribution of the total delay obeys the product of convolution of the distributions of delay at each of the three stages. Because there is a great deal of mixing of flows in the network and in accordance with what has already been pointed out in section 2.1.3 and observed by many authors (Ref.11), we can assume that the flows within the network follow a Poisson distribution. This therefore entails deriving the product of convolution of three queues $M/D/1$.

This is an easy process, using the approximate formula below (ref.5) which is very accurate even for low values of ρ :

$$P(> x) = -\frac{1-\rho}{\ln(\rho)} e^{-(1-\rho-\ln(\rho))x} \quad (15)$$

The product of convolution for three queues such that : $P_i(=x) = \alpha_i e^{-a_i x}$ ($i = 1, 2, 3$) is easily obtained from the Laplace Transform :

$$P_{(3)}^*(s) = P_1^*(s) \cdot P_2^*(s) \cdot P_3^*(s) = \frac{\alpha_1 \alpha_2 \alpha_3}{(s + a_1)(s + a_2)(s + a_3)} \quad (16)$$

From formula (16) we can deduce : $P_{(3)}(> x) = K_1 e^{-a_1 x} + K_2 e^{-a_2 x} + K_3 e^{-a_3 x}$ (17)

$$\begin{aligned} \text{where : } K_1 &= \frac{\alpha_1 \alpha_2 \alpha_3}{a_1(a_2 - a_1)(a_3 - a_1)}, \quad K_2 = \frac{\alpha_1 \alpha_2 \alpha_3}{a_2(a_1 - a_2)(a_3 - a_2)} \\ K_3 &= \frac{\alpha_1 \alpha_2 \alpha_3}{a_3(a_1 - a_3)(a_2 - a_3)}, \quad a_i = 1 - \rho_i - \ln(\rho_i) \text{ and} \\ \alpha_i &= -a_i \frac{1 - \rho_i}{\ln(\rho_i)} \end{aligned}$$

NUMERICAL APPLICATION

From the results in the sections above, we can consider a maximum cell load of 0.9 on the ATM links coming into the network. This value is a conservative one with respect to the value obtained in section 1.3. Given for instance an internal structure with an expansion of three, the values of ρ to be allowed for at each stage are : $\rho_1 = 0.3, \rho_2 = 0.3, \rho_3 = 0.9$.

This then gives, using (17), $P_{(3)}(> x) = 10^{-10}$ for $x=111$ cells in the system (a value due to the preponderance of the third stage). The maximum CDV is therefore $76 \mu s$ ($111 \times 682 \text{ nsec}$), which is fully compatible with the real-time constraints of 64 Kbit/s services, the ATM bursty traffics being as for them penalised at call acceptance level and blocking at the input of the network. As a consequence, CDV can not be considered in this case as a real constraint, call blocking probability remaining the preponderant factor for the network dimensioning.

Furthermore it is easy to verify that even with a load of 0.9 at each stage the maximum CDV is the same with a probability of $3 \cdot 10^{-10}$. That means that under the constraint of a negligible internal call rejecting probability, expansion is even no necessary. This is particularly interesting for matrices connecting narrow-band services for which it is easy to get very low internal blocking probability without any expansion.

CONCLUSION

In this study we have evaluated the performance of a switching matrix based on the ATM composite technique. We have established the formulae which enable the network to be dimensioned for 64 Kbit/s services, and also for Broad-Band services.

The main contributions of this paper are thus, the application of Brockmeyer's work on overflow systems, which enabled us to give the exact distribution of the number of cells required and the loss probability for 64 Kbit/s services, and second, the determination of a very accurate formula for the calculation of the blocking probability for Broad-Band services which yields quite a good network dimensioning rule and accurate call acceptance algorithms.

From the point of view of traffic flow, the results obtained show the efficiency of a Clos type structured network. This kind of network without any blocking, or with a negligible one, at the VC and VP levels, is entirely effective and, furthermore, small capacities of the queues reserved to each elementary switching matrix are sufficient to guarantee a good service quality (crossing delay and loss probability).

We therefore consider that the formulae established can serve as a basis for drawing up ATM traffic control procedures. Indeed, traffic characteristics such as peak and mean bit rates combined with enumeration systems of calls provide the means of definition of engineering tables and call acceptance rules which enable to guarantee a good quality of service.

In a future work we shall study the dimensioning of the network when allowing very low call rejection probability within the ATM core (quasi nonblocking network instead of strictly nonblocking network), while maintaining negligible multiplex saturation probability, such as suggested in section 2.2.

REFERENCES

- [1] SPANKE R.A., MARK ADRIAN J. ATM composite cell switching for DSO digital switches. *ISS 95 vol1*, pp 268-272
- [2] CLOS C. A Study of Non Blocking Switching Network . *BSTJ Vol.32 1953* pp 406-424.
- [3] BROCKMEYER E. The simple overflow problem in the theory of telephone traffic. *Teleteknik*, 5, N°4, pp 361-374, Dec. 1954.
- [4] SAITO T., INOSE H., HAYASHI S. Evaluation of traffic carrying capability in one-stage and two -stage time-division networks handling data with a variety of speed classes. *ITC 9*, Oct. 1979.
- [5] FICHE G., LÖRCHER W., OGER F. & F., VEYLAND R. Study of multiplexing for ATM traffic sources. *ITC 14, Vol 1a*, pp 441-452, June 1994.
- [6] ANICK D., MITRA D., SONDDHI M. M. Stochastic Theory of a Data-Handling System with Multiple Sources. *The BELL Technical Journal*, Vol.61, N°8, pp 1871-1894, Dec. 1981.
- [7] YANG T., TSANG D. H. K. A Novel Approach to Estimating the Cell Loss Probability in an ATM Multiplexer Loaded with Homogeneous ON-OFF Sources. *IEEE Transactions on COMMUNICATIONS*, Vol.43, N°1, pp 117-126, Jan. 1995.
- [8] GUIBERT J. Overflow probability upper bound for heterogeneous fluid queues handling general ON-OFF Sources. *ITC 14, Vol 1a*, pp 65-74, June 1994.
- [9] BAIOCCHI A., BLEFARI-MELAZZI N., ROVERI A., SALVATORE F. Stochastic Fluid Analysis of an ATM Multiplexer Loaded with Heterogeneous ON-OFF Sources : an Effective Computational Approach. *INFOCOM '92*, pp 405-414, 1992.
- [10] WOJCIECH KABACINSKI On Non blocking Switching Networks for multirate connections. *ITC 13, Vol 14*, pp885-889, June 1991.
- [11] MONTAGNA S., PAGLINO R., MEYER J.F. Delay Performance of a Multistage ATM Switching Network. *ITC 14, Vol 1a*, pp 623-634, June 1994.

BIBLIOGRAPHY

Georges Fiche is Technical Performance Manager for Alcatel CIT, based in Lannion France, and Technical Coordinator for Performance Standardization for Alcatel Telecom. His fields of activity include performance evaluation for switching equipment, covering both trafficability and dependability, as well as performance modeling for new communication technologies. Claude Le Palud is member of the Technical Performance Study Group based in Lannion. His fields of activity include traffic performance modeling as well as performance measurements for switching equipment and new communication technologies. Stéphane Rouillard is a student at IMA (Institut de Mathématiques Appliquées), Angers France. He did graduate work at Alcatel Lannion, in the Technical Performance Study Group.

A Study of the Fairness of the Fast Reservation Protocol*

Llorenç Cerdà, Jorge García and Olga Casals

Polytechnic University of Catalonia

Computer Architecture Department

c/ Gran Capitan, Modulo D6, E-08071 Barcelona, Spain

tel : + 34 3 4016798, fax : + 34 3 4017055,

e-mail : llorenc@ac.upc.es

Abstract

Fast Reservation Protocol (FRP) is a Traffic Control scheme intended to multiplex bursty data sources. In this paper we focus on the analysis of the FRP when different sources are multiplexed together in order to study the fairness of the protocol. We present two analytical models to analyse the case in which a set of identical sources is multiplexed with another one of higher rate. Analytical results are compared with simulation results.

1 INTRODUCTION

In order to efficiently multiplex data transfers and LAN-LAN interconnection on the ATM B-ISDN, an in-call bandwidth negotiation called Fast Reservation Protocol (FRP) has been proposed, Boyer (1992). FRP is a kind of Connection Acceptance Control at burst level, that is, when a source wants to transmit a burst it is accepted or blocked depending on the available bandwidth within the link. When a burst is blocked successive reattempts are made until it is accepted. Although the FRP it is not a new proposal, it is still a hot topic because recently it has been included in the ITU-T 371 recommendation to support the ATM Block Transfer Capability.

Performance of an FRP connection is therefore measured in terms of its Burst Blocking Probability (BP) and its Blocking Time (BT, i.e. the time that a blocked burst has to wait until it is eventually accepted). Performance studies of FRP and related protocols have been carried out by several authors, Boyer (1992), Enssle (1994), Suzuki (1992), Bernstein (1994). In those studies however, a set of identical sources is used to model the protocol behaviour. When sources with different parameters (PCR and/or burst duration) are multiplexed together, it is foreseeable that each source type will get a different BP and BT. In this paper we focus on the analysis of the FRP fairness when different sources are multiplexed. We use the term fairness in the sense of discrepancy between BP and

*This work was supported by the Ministry of Education of Spain under grant TIC94-1512-CE

BT values of different source types. Being all equal, the network would have a fair burst access.

We assume an ON-OFF model for the data sources with exponential ON and OFF time distribution (burst-silence model). In order to assess the burst blocking probability of the sources, two approximations of the protocol are considered. In the first approach we assume that the time between reattempts is zero. With different types of sources this case leads to a Markov chain that does not have a product form solution, so we analyze the simple situation in which a set of identical sources are multiplexed with another source of a higher rate.

In a second approach we consider that the reattempt time and OFF time are identically distributed. This assumption leads to a Markov chain with a simple product form solution even when considering different source types.

In the first approach the time that a burst has to wait when it is blocked until it is accepted is also evaluated. Analytical results are compared with simulation results.

2 OVERVIEW OF THE FRP PROTOCOL

The FRP is described in Boyer (1992). Two variants of the protocol have been proposed. The first, called FRP with Delayed Transmission (FRP/DT), is intended to multiplex the so called Stepwise Variable Bit Rate Sources. These sources are expected to have a stepwise need of bandwidth. However there is a restriction on the sources which must tolerate a delay in the negotiation of an increase of bandwidth. Many data communications are typical examples of such sources.

Basically the FRP/DT works as follows. When a source wants an increase of bandwidth (for example, when it wants to transfer a burst), it sends a Request to the so called FRP Control Unit, situated at the ingress node. This Request is forwarded to the first switching element of the link, which checks whether it can allocate the increase of bandwidth or not. If it has enough bandwidth, the Request is forwarded to the next switching element and so on until it reaches the egress node. Eventually the egress node will send an acknowledgment back to the FRP Unit and the source will be allowed to transfer the burst. The time passed from the FRP Unit sending the request until receiving the acknowledgment is called the Round Trip Time. Note that during this time the switching elements have allocated bandwidth for the source, but the transmission has not started yet. Therefore this time is an overhead introduced by the protocol.

If a switching element is not able to allocate the requested increase of bandwidth, it discards the Request, and by a time-out mechanism the allocated resources are reset to their previous state. In this case the FRP Unit makes successive reattempts until the increase of bandwidth is accepted. The source indicates the FRP Unit when an accepted burst is already transferred in order to release the allocated bandwidth.

The other variant, called FRP with immediate transmission (FRT/IT), is intended for sources more sensitive to a time delay. In this case the source transfers the burst immediately after the reservation request. If the reservation fails in any of the nodes, the whole burst is discarded.

3 MODEL DESCRIPTION AND ANALYSIS

In our analysis we consider an isolated node. We assume an ON-OFF model for the data sources with exponential ON and OFF time distribution (burst-silence model). The parameters of the sources are the bitrate within a burst period Λ ; the mean burst duration t^{on} and the mean silence duration t^{off} .

In the model N_l identical sources (we will refer to them as *ltype* sources) with parameters Λ_l , t_l^{on} and t_l^{off} , are multiplexed with another different source (we will refer to it as *htype* source) with parameters Λ_h , t_h^{on} and t_h^{off} .

Being all time intervals exponentially distributed, the activation rate α of a source is given by $\alpha = 1/t^{off}$. Let the service time be the time that a node allocates bandwidth for a non blocked source. Clearly, for the FRP/IT the mean service time is the mean burst duration t^{on} . For the FRP/DT a non blocked source has to wait a deterministic time equal to the round trip time t_{rt} before transferring a burst, so the mean service time is given by $t^{on} + t_{rt}$. However, in this paper we do not study the influence of the round trip time, so we will assume it to be zero. Therefore the service rate μ of a source is given by $\mu = 1/t^{on}$. Assuming $t_{rt} = 0$ our model makes no distinction between the FRP/DT and FRP/IT. Refer to Enssle (1994) for a contrast of both variants of the protocol.

3.1 Approximation by zero time between reattempts

In this approximation we suppose that when a burst is blocked, the time between the successive requests that are made until the burst is accepted is zero. This is equivalent to considering a blocked burst being kept in a queue until there is enough bandwidth left by the other sources in the link.

Let K_1 be the maximum number of *ltype* sources that can be simultaneously transferring a burst without exceeding the link capacity, when the *htype* source is also transferring a burst. Let K_2 be the same, but when the *htype* source is silent or blocked. Let us further suppose that the *htype* source transmits at a higher rate than the *ltype* source such that $K_2 > K_1 + 1$. In this case when an *ltype* and *htype* sources are blocked, the *ltype* source will be accepted first (i.e. the *htype* source does not see a FIFO queue). Clearly, if the link capacity is C

$$K_1 = \left\lfloor \frac{C - \Lambda_h}{\Lambda_l} \right\rfloor \quad (1)$$

$$K_2 = \left\lfloor \frac{C}{\Lambda_l} \right\rfloor \quad (2)$$

With these assumptions an isolated node can be described by the Markov chain of figure 1 with state space $\{(i, j) : i = 0, 1, 2 ; 0 \leq j \leq N_l\}$, where j is the number of *ltype* active sources (transferring or blocked) while the *htype* source is silent ($i = 0$), transferring a burst ($i = 1$) or blocked ($i = 2$). This Markov chain does not have a product form solution for the stationary probabilities π_{ij} , so they have to be calculated numerically solving the global balance equations.

The *ltype* and *htype* source blocking probability (P_l and P_h) can be obtained from the

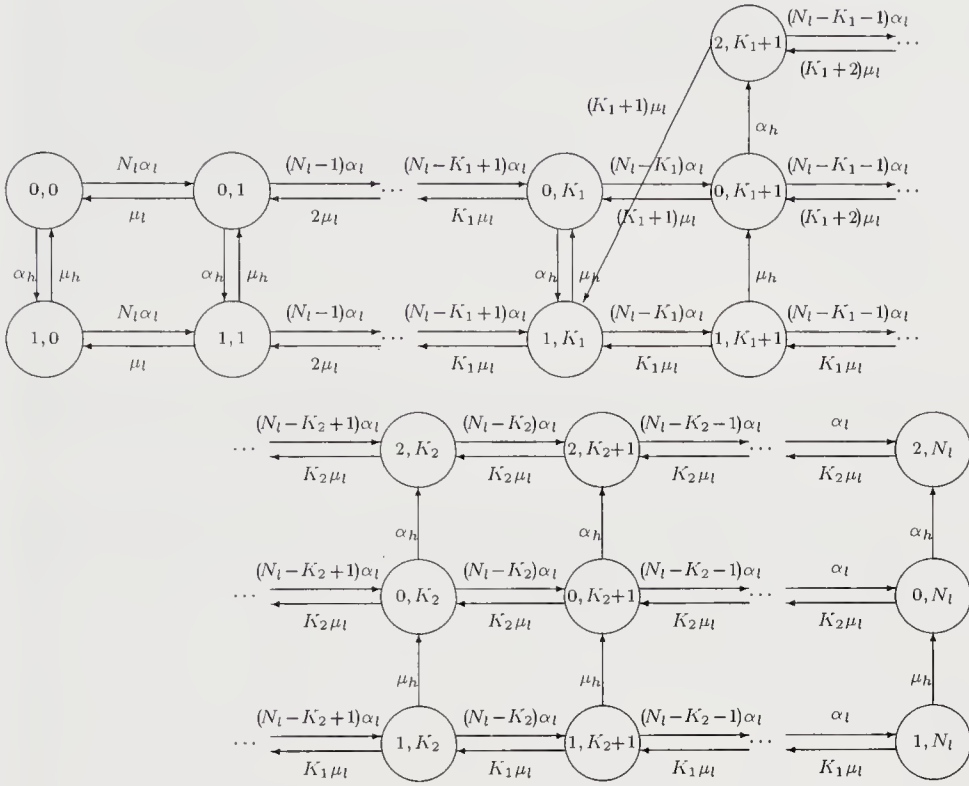


Figure 1 State-transition diagram assuming zero time between reattempts

stationary probabilities π_{ij} . The blocking probability is given by the probability that an arriving burst is blocked, divided by the probability of a burst arrival. Thus

$$P_h = \frac{\sum_{j=K_1+1}^{N_l} \pi_{0j}}{\sum_{j=0}^{N_l} \pi_{0j}} \quad (3)$$

$$P_l = \frac{\sum_{j=K_2}^{N_l-1} (N_l - j) \pi_{0j} + \sum_{j=K_1}^{N_l-1} (N_l - j) \pi_{1j} + \sum_{j=K_2}^{N_l-1} (N_l - j) \pi_{2j}}{\sum_{j=0}^{N_l-1} (N_l - j) \pi_{0j} + \sum_{j=0}^{N_l-1} (N_l - j) \pi_{1j} + \sum_{j=K_1+1}^{N_l-1} (N_l - j) \pi_{2j}} \quad (4)$$

3.2 Approximation by identically reattempt and OFF time distribution

In this approximation we assume that when a burst is blocked, the time between the successive requests that are made until the burst is accepted is exponentially distributed

with a mean equal to the OFF time distribution, i.e. we assume an identically reattempt and OFF time distribution. This is equivalent to considering that a blocked burst is lost.

Let K_1 and K_2 be the same as in the previous section. Because a blocked burst can be considered as lost, with this approach an isolated node can be described by the Markov chain with state space $\{(i, j) : i = 0, 1 ; 0 \leq j \leq K_2\}$ of figure 2, where j is the number of ltype sources transferring a burst while the htype source is silent ($i = 0$) or transferring a burst ($i = 1$). The stationary probabilities π_{ij} of the Markov chain has a straightforward product form solution given by

$$\pi_{ij} = \frac{1}{G} \binom{N_l}{j} \rho_h^i \rho_l^j \quad (5)$$

where G is the normalization constant, $\rho_l = \mu_l/\alpha_l$ and $\rho_h = \mu_h/\alpha_h$. We note that considering more than one htype source or even considering more than two types of sources, a product form solution would still apply.

In this model we make no distinction between a burst or a reattempt arrival. So we calculate the blocking probability as the probability that a burst or a reattempt arrival is blocked, divided by the probability of a burst or a reattempt arrival. Such blocking probability for the ltype and htype sources (P_l and P_h) is given by

$$P_h = \frac{\sum_{j=K_1+1}^{K_2} \pi_{0j}}{\sum_{j=0}^{K_2} \pi_{0j}} \quad (6)$$

$$P_l = \frac{(N_l - K_2)\pi_{0K_2} + (N_l - K_1)\pi_{1K_1}}{\sum_{j=0}^{K_2} (N_l - j)\pi_{0j} + \sum_{j=0}^{K_1} (N_l - j)\pi_{1j}} \quad (7)$$

Note that in the previous section we do not count the reattempts to calculate the blocking probability (considering a zero time between reattempts implies considering ∞ reattempts after a blocked burst). If the reattempt time is not zero, the following relation applies for the P_l^{init} and P_l^{total} blocking probabilities of an ltype source, calculated counting and not counting the reattempts respectively. Let \bar{r}_l be the mean number of reattempts that a blocked burst of an ltype source do until it is accepted. It can be derived that

$$P_l^{init} = \frac{P_l^{total}}{\bar{r}_l (1 - P_l^{total})} \quad (8)$$

Obviously, an analogous relation holds for the htype source. If the blocking probability is small and the reattempt time is high enough such $\bar{r} \approx 1$ (i.e. a blocked burst is almost always accepted at the first reattempt), $P^{init} \approx P^{total}$. These conditions are foreseeable in the approximation by identically reattempt and OFF time distribution. So, this approximation can be used to asses P_h^{init} and P_l^{init} from the probabilities calculated with equations 6 and 7.

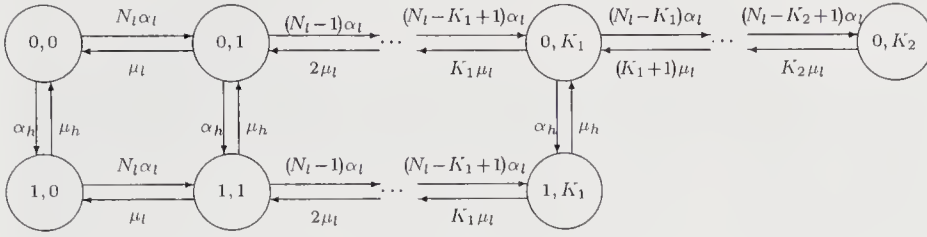


Figure 2 State-transition diagram assuming identically reattempt and OFF time distribution

3.3 Blocking time in the approximation by zero time between reattempts

In this section we calculate the time that an arriving burst that is blocked has to wait until it is eventually accepted (we refer to it as blocking time). We calculate this time assuming the approximation by zero time between reattempts, so the referred states are those of figure 1. We do not use the approximation by identically reattempt and OFF time distribution to assess the blocking time, because in general it would be inaccurate.

Let T_h and T_l be the blocking time of an htype source and ltype source respectively. Let $B_{ij} = (i, j)$ be entering state resulting from the blocking transition. Clearly

$$P(T_h \leq x) = \sum_{j=K_1+1}^{N_l} P(T_h \leq x | B_{2j}) P(B_{2j}), \quad (9)$$

$$P(B_{2j}) = \frac{\pi_{0j}}{\sum_{k=K_1+1}^{N_l} \pi_{0k}} \quad (10)$$

and

$$P(T_l \leq x) = \sum_{\forall B_{ij}} P(T_l \leq x | B_{ij}) P(B_{ij}), \quad (11)$$

$$P(B_{ij}) = \frac{(N_l - j + 1) \pi_{ij-1}}{\sum_{k=K_2}^{N_l-1} (N_l - k) \pi_{0k} + \sum_{k=K_1}^{N_l-1} (N_l - k) \pi_{1k} + \sum_{k=K_2}^{N_l-1} (N_l - k) \pi_{2k}} \quad (12)$$

$P(T_h \leq x | B_{ij})$ is the distribution of the time that a blocked burst of an htype source has to wait until it is accepted, when the entering state in the blocking transition is B_{ij} . $P(T_l \leq x | B_{2j})$ is the same for an ltype source. Formulas for this probabilities are derived in appendixes 1 and 2.

4 RESULTS

In this section we present a numerical study of the FRP fairness using the models described above. We evaluate the fairness of the protocol in terms of the burst blocking probability and the mean blocking time. Blocking time is specially important using the FRP/IT scheme in which the sources are supposed to be time sensitive. We also compare analytical and simulation results.

Figures 3 and 4 (model parameters are summarized in Table 2) plot the blocking probability and the mean blocking time of the two source types considered, when the ltype source varies the mean burst duration (i.e. the mean ON time t_h^{on})[†]. Varying t_h^{on} from 0 (the source is always silent) to ∞ (the source is always active), blocking probability of the ltype sources will increase between the one obtained when sharing a link of capacity varying from C to $C - \Lambda_h$. Figure 3 shows that the blocking probability of the ltype sources increases within these limits, while the blocking probability of the htype source remains constant. The blocking probability is assessed using the approximation by zero time between reattempts (section 3.1)[‡], and the approximation by identical reattempt and OFF time distribution (section 3.2).

Figures 5 and 6 plot the blocking probability and the mean blocking time of the two source types, when the htype source varies the bitrate within a burst period. Each time that the htype source bitrate reaches a multiple of the ltype source bitrate, there is a decrement on the maximum number of sources that can be simultaneously transferring a burst. This causes an increasing step on the blocking probability and the blocking time.

Table 1 compares analytical and simulation results (given with 95% confidence intervals). To calculate the blocking probabilities in the simulation, the reattempts have been not counted in order to compare with the approximation by zero time between reattempts (these probabilities are referred to as “init.” in the table), and have been counted to compare with the approximation by identically reattempt and OFF time dist. (referred to as “tot.” in the table, cfr. section 3.2). Increasing the reattempt time decreases the blocking probability. So, the first approximation can be considered as an upper bound for the “init.” probabilities, and, for a reattempt time lower than the mean OFF time, the second approximation can be considered as a lower bound for the “tot.” probabilities.

A deterministic and an exponentially distributed reattempt time has been considered in the simulation. It can be seen that the exponentially distributed approximation for the reattempt time gives accurate results for the blocking probabilities, but the blocking time. Simulation results show that the mean blocking time increases rapidly with increasing the reattempt time.

5 CONCLUSIONS

We have analyzed the behaviour of the FRP when different source types are multiplexed together. We have considered the case in which a set of identical sources is multiplexed with another one of higher bitrate. To assess the blocking probability we have considered

[†]We note that the burstiness, defined as $b = (t^{on} + t^{off})/t^{on}$ is a decreasing function with increasing t^{on} .

[‡]To calculate the stationary probabilities using this approximation, we have solved the global balance equations using a Gaussian elimination method

Table 1 Comparison of analytical and simulation results

	Analytical		Simulation					
	Zero time between reatt.	Id. reatt. and OFF time dist.	Reattempt time					
			5 ms		20 ms		50 ms	
			Exp. dist.	Det.	Exp. dist.	Det.	Exp. dist.	Det.
P_h init.	$7.63 \cdot 10^{-3}$		$7.00 \cdot 10^{-3}$ $\pm 2.83 \cdot 10^{-4}$	$6.86 \cdot 10^{-3}$ $\pm 4.30 \cdot 10^{-4}$	$7.31 \cdot 10^{-3}$ $\pm 5.33 \cdot 10^{-4}$	$7.28 \cdot 10^{-3}$ $\pm 6.46 \cdot 10^{-4}$	$7.00 \cdot 10^{-3}$ $\pm 2.49 \cdot 10^{-4}$	$7.16 \cdot 10^{-3}$ $\pm 8.72 \cdot 10^{-4}$
P_l init.	$8.22 \cdot 10^{-4}$		$5.76 \cdot 10^{-4}$ $\pm 1.19 \cdot 10^{-5}$	$5.92 \cdot 10^{-4}$ $\pm 1.78 \cdot 10^{-5}$	$5.13 \cdot 10^{-4}$ $\pm 4.89 \cdot 10^{-5}$	$4.98 \cdot 10^{-4}$ $\pm 5.28 \cdot 10^{-5}$	$4.28 \cdot 10^{-4}$ $\pm 1.10 \cdot 10^{-5}$	$4.31 \cdot 10^{-4}$ $\pm 4.37 \cdot 10^{-5}$
P_h tot.		$7.57 \cdot 10^{-3}$	$29.7 \cdot 10^{-3}$ $\pm 1.65 \cdot 10^{-3}$	$25.6 \cdot 10^{-3}$ $\pm 1.75 \cdot 10^{-3}$	$14.4 \cdot 10^{-3}$ $\pm 1.33 \cdot 10^{-3}$	$12.1 \cdot 10^{-3}$ $\pm 1.34 \cdot 10^{-3}$	$10.1 \cdot 10^{-3}$ $\pm 0.35 \cdot 10^{-3}$	$8.52 \cdot 10^{-3}$ $\pm 1.12 \cdot 10^{-3}$
P_l tot.		$4.15 \cdot 10^{-4}$	$13.9 \cdot 10^{-4}$ $\pm 2.32 \cdot 10^{-5}$	$12.4 \cdot 10^{-4}$ $\pm 4.29 \cdot 10^{-5}$	$6.80 \cdot 10^{-4}$ $\pm 6.36 \cdot 10^{-5}$	$5.90 \cdot 10^{-4}$ $\pm 6.57 \cdot 10^{-5}$	$4.90 \cdot 10^{-4}$ $\pm 1.09 \cdot 10^{-5}$	$4.47 \cdot 10^{-4}$ $\pm 4.63 \cdot 10^{-5}$
T_h (ms)	15.24		22.0 ± 0.32	19.13 ± 0.48	40.8 ± 0.95	33.6 ± 0.88	73.2 ± 1.23	59.9 ± 0.92
T_l (ms)	6.822		12.3 ± 0.07	10.51 ± 0.16	28.0 ± 0.24	23.7 ± 0.22	57.5 ± 0.41	51.8 ± 0.16

two approximations. In the first one we assume that the reattempt time is zero and in the second one we assume that it is identically distributed to the OFF time. To calculate the stationary state probabilities with the first approach the balanced global equations have to be solved, while in the second approach they have a simple product form solution. We have also calculated the mean blocking time assuming the first approach.

The numerical study shows that there are not big differences between the blocking probabilities obtained with both approximations. The approximation of identical reattempt and OFF time distribution gives a much simple way to compute the blocking probabilities and can be easily extended to more than one htype source or even more than two types of sources.

The results also show that when multiplexing different type of sources, blocking probability and blocking time depend on the source parameters. This can be interpreted as a lack of fairness, in the sense that they will have a different burst access. It is actually seen that an increase on the bitrate or the mean burst duration of a connection can result in a considerably increase of the blocking probability and blocking time of the other connections.

Recently the ATM Block Transfer Capability (ABT) with two variants ABT/DT and ABT/IT based on the FRP/DT and FRP/IT respectively have been defined, ITU (1995). In this recommendation a Block level QoS commitment is defined in which a reservation request should be accepted by the network within finite time limits (ABT/DT), or with a specified block discard probability (ABT/IT), as long as blocks of the connection are conforming to the specified Sustainable Cell Rate. These QoS parameters can be easily derived from the blocking probability and blocking time parameters we have measured (we note that such ITU recommendation is subsequent to the study carried out in this paper).

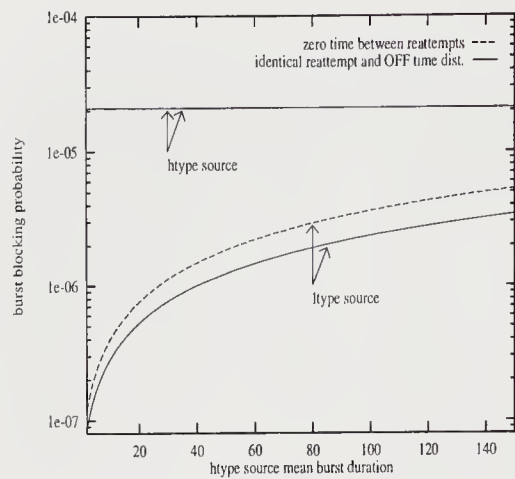


Figure 3 Influence of the htype source mean burst duration on the blocking probability

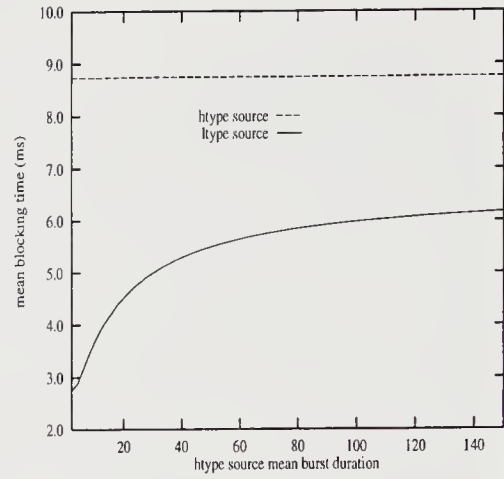


Figure 4 Influence of the htype source mean burst duration on the mean blocking time

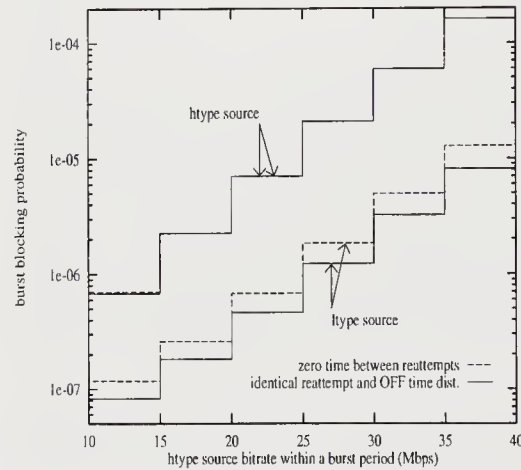


Figure 5 Influence of the htype source bitrate within a burst period on the blocking probability

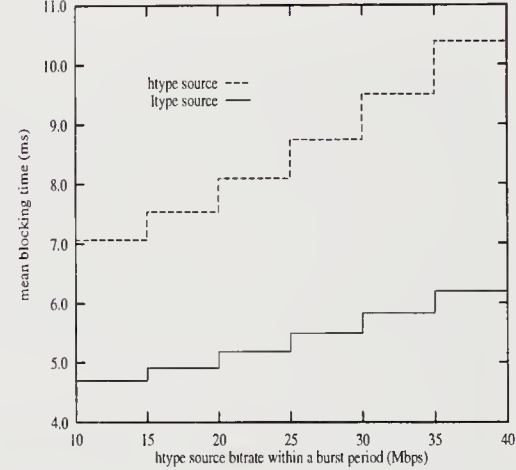


Figure 6 Influence of the htype source bitrate within a burst period on the mean blocking time

Table 2 Source parameters

link	htype source				ltype source				
	capacity	bitrate	t_h^{on}	t_h^{off}	burst-	bitrate	t_l^{on}	t_l^{off}	num. of
	150 Mbps	(Mbps)	(ms)	(ms)	iness	(Mbps)	(ms)	(ms)	sources
fig. 3 & 4	30	0~150	1400	∞	~ 10.3	5	100	1400	15
fig. 5 & 6	10~40	50	1400	29		5	100	1400	15
table 1	30	50	900	19		5	100	900	10

APPENDIX 1

In this appendix we derive $P(T_h \leq x | B_{2j})$, $j \in \{K_1 + 1, \dots, N_l\}$ of expression 9. If an htype and ltype sources are blocked, the ltype source will be accepted first (i.e. the htype source does not see a FIFO queue), so $P(T_h \leq x | B_{2j})$ is the distribution of the first passage time from the blocking state B_{2j} to the non blocking state $(1, K_1)$. To calculate this probability we follow the method described in Neuts (1989). We are only concerned about the states $(1, K_1), (2, K_1 + 1), \dots, (2, N_l)$ so to simplify the notation we will refer to them as E_j , $j = K_1, K_1 + 1, \dots, N_l$. We define the following events

$$\begin{aligned} T(j, j-r) &= \text{first passage time from state } E_j \text{ to state } E_{j-r} \\ V(j, j-r) &= \text{number of transitions involved in } T(j, j-r) \end{aligned}$$

and their joint probability $G_j^{(r)}(x, k) = P\{T(j, j-r) \leq x, V(j, j-r) = k\}$. The probability we are looking for is given by

$$P(T_h \leq x | B_{2j}) = \sum_{k=1}^{\infty} G_j^{(j-K_1)}(x, k) \quad (13)$$

We now compute $G_j^{(r)}(x, k)$. To simplify the notation, in case of one state transition we will write $G_j(x, k) = G_j^{(1)}(x, k)$. Let q_j^+ be the transition rate from the state E_j to the state E_{j+1} ; q_j^- the transition rate from the state E_j to the state E_{j-1} ; and q_j the self state transition rate (cfr. figure 1).

$$\begin{aligned} q_j^+ &= (N_l - j) \alpha_l \\ q_j^- &= \begin{cases} j \mu_l, & j \leq K_2 \\ K_2 \mu_l, & j > K_2 \end{cases} \\ q_j &= q_j^+ + q_j^- \end{aligned}$$

We define the one state forward and backward transition probability $A_j^-(x) = P\{T(j, j-1) \leq x, V(j, j-1) = 1\}$ and $A_j^+(x) = P\{T(j, j+1) \leq x, V(j, j+1) = 1\}$. We have

$$\begin{aligned} A_j^-(x) &= (1 - e^{-q_j x}) \frac{q_j^-}{q_j}, \quad j = K_1 + 1, \dots, N_l \\ A_j^+(x) &= (1 - e^{-q_j x}) \frac{q_j^+}{q_j}, \quad j = K_1, \dots, N_l - 1 \end{aligned} \quad (14)$$

yielding

$$G_j(x, k) = \begin{cases} A_j^-(x), & k = 1 \\ 0, & k = 2n \\ \sum_{l=1}^n A_j^+(\cdot) * G_{j+1}(\cdot, 2(n-l)+1) * G_j(x, 2l-1), & k = 2n+1 \end{cases} \quad (15)$$

and

$$G_j^{(r)}(x, k) = \sum_{k_1 + \dots + k_r = k} G_j(\cdot, k_1) * G_{j-1}(\cdot, k_1) * \dots * G_{j-r+1}(x, k_r) \quad (16)$$

where $*$ is the convolution of the distribution functions (i.e. $F_1(\cdot) * F_2(x) = \int_{-\infty}^{\infty} F_1(x - \lambda) dF_2(\lambda)$)

From 15 we derive the following recursive equation for the joint transform

$$\tilde{G}_j(s, z) = \sum_{k=0}^{\infty} \int_0^{\infty} e^{-sx} z^k dG_j(x, k)$$

$$\tilde{G}_j(s, z) = \frac{z A_j^-(s)}{1 - z A_j^+(s) \tilde{G}_{j+1}(s, z)} \quad (17)$$

and

$$\begin{aligned} \tilde{G}_j^{(r)}(s, z) &= \sum_{k=0}^{\infty} z^k \sum_{k_1 + \dots + k_r = k} G_j(s, k_1) G_{j-1}(s, k_1) \dots G_{j-r+1}(s, k_r) = \\ &\tilde{G}_j(s, z) \tilde{G}_{j-1}(s, z) \dots \tilde{G}_{j-r+1}(s, z) \end{aligned} \quad (18)$$

where $A_j^-(s)$, $A_j^+(s)$ and $G_j(s, k)$ are the Laplace-Stieltjes transform of the distribution functions (i.e. $F(s) = \int_0^{\infty} e^{-sx} dF(x)$). From 14 we obtain

$$\begin{aligned} A_j^-(s) &= \frac{q_j^-}{s + q_j}, \quad j = K_1 + 1, \dots, N_l \\ A_j^+(s) &= \frac{q_j^+}{s + q_j}, \quad j = K_1, \dots, N_l - 1 \end{aligned} \quad (19)$$

Substitution into 17 yields

$$\tilde{G}_j(s, z) = \frac{z q_j^-}{s + q_j^+ - z q_j^- \tilde{G}_{j+1}(s, z)}, \quad j = K_1 + 1, \dots, N_l - 1 \quad (20)$$

$$\tilde{G}_{N_l}(s, z) = z \frac{q_{N_l}^-}{s + q_{N_l}^-} \quad (21)$$

Substituting recursively 21 into 20, and then into 18 we obtain $\tilde{G}_j^{(r)}(s, z)$, $j = K_1 + 1, \dots, N_l$. Finally, from 13 we see that $G_j^{(j-K_1)}(s, z)_{z=1}$ is the Laplace-Stieltjes transform of $P(T_h \leq x | B_{2j})$. Inverting it and substituting into 9 we obtain the distribution of the blocking time T_h . This is rather arduous, but from the previous equations we can derive a straightforward formula for the mean blocking time \bar{T}_h .

Let us define $\bar{T}_j^{(j-K_1)} = E[x | B_j]$, i.e. the mean first passage time from the state E_j to the state E_{K_1} . We also define \bar{T}_j as the mean first passage time from the state E_j to the state E_{j-1} . Clearly $\bar{T}_j^{(j-K_1)} = - \frac{\partial}{\partial s} \tilde{G}_j^{(j-K_1)}(s, z) \Big|_{s=0, z=1}$, $\bar{T}_j = - \frac{\partial}{\partial s} \tilde{G}_j(s, z) \Big|_{s=0, z=1}$. From 20, 21

and 18, and since $\tilde{G}'_j(s, z)_{s=0, z=1} = 1$ we obtain

$$\bar{T}_j = \frac{1 + q_j^+ \bar{T}_{j+1}}{q_j^-} \quad (22)$$

$$\bar{T}_{N_l} = \frac{1}{q_{N_l}^-} \quad (23)$$

$$\bar{T}_j^{(j-K_1)} = \sum_{k=K_1+1}^j \bar{T}_k \quad (24)$$

Substituting recursively 23 into 22, and then into 24 we compute $\bar{T}_j^{(j-K_1)}$ and finally from 9 we obtain the mean blocking time

$$\bar{T}_h = \sum_{j=K_1+1}^{N_l} \bar{T}_j^{(j-K_1)} P(B_{2j}) \quad (25)$$

APPENDIX 2

In this appendix we derive the $P(T_l \leq x | B_{ij})$ of the expression 11. To calculate this probability we consider the following cases:

1. $B_{ij} \in \{(2, K_2 + 1), \dots, (2, N_l), (0, K_2 + 1), \dots, (0, N_l)\}$

In this case the ltype source is blocked while the htype source is silent or blocked. Being $B_{ij} = (i, j)$ the state resulting from the blocking transition, the source will find $j - K_2 - 1$ ltype sources already blocked and it will have to wait until $j - K_2$ ltype sources are served (we say that a source is served when one of the K_2 bursts being transferred ends). Let $S_l^{(n)}$ be the service time of n ltype sources and $F_{S_l}^{(n)}(x)$ its distribution. Clearly $F_{S_l}^{(1)}(x) = 1 - e^{-K_2 \mu_l x}$ and $F_{S_l}^{(n)}(x) = F_{S_l}^{(1)}(\cdot) * \dots * F_{S_l}^{(1)}(x)$. Let $F_{T_l | B_{ij}}(s)$ be the Laplace-Stieltjes transform of $P(T_l \leq x | B_{ij})$. We have

$$F_{T_l | B_{ij}}(s) = F_{S_l}^{(j-K_2)}(s) = \frac{(K_2 \mu_l)^{j-K_2}}{(s + K_2 \mu_l)^{j-K_2}} \quad (26)$$

2. $B_{ij} \in \{(1, K_1 + 1), \dots, (1, K_2)\}$

In this case the ltype source is blocked while the htype source is transferring a burst, but the maximum of active ltype sources is K_2 . Being $B_{ij} = (i, j)$ the state resulting from the blocking transition, the source will find $j - K_1 - 1$ ltype sources already blocked. So to be accepted it will have to wait until the htype source is served or until $j - K_1$ ltype sources are served. Let S_h be the service time of the htype source and $S_l^{(n)}$ the service time of n ltype sources. Let $F_{S_h}(x)$ and $F_{S_l}^{(n)}(x)$ be their distribution. Clearly $F_{S_h}(x) = 1 - e^{-\mu_h x}$ and $F_{S_l}^{(n)}(x)$ is the same as in the previous case, but changing K_2 for K_1 . We have

$$\begin{aligned} P(T_l \leq x | B_{ij}) &= 1 - P(S_h > x) P(S_l^{(j-K_1)} > x) = \\ &= 1 - (1 - F_{S_h}(x))(1 - F_{S_l}^{(j-K_1)}(x)) = 1 - (1 - F_{S_l}^{(j-K_1)}(x)) e^{-\mu_h x} \end{aligned} \quad (27)$$

the Laplace-Stieltjes of the previous equation is

$$F_{T_l|B_{ij}}(s) = s \int_0^\infty \left(e^{-sx} - (1 - F_{S_l}^{(j-K_1)}(x)) e^{-(s+\mu_l)x} \right) dx =$$

$$1 - \frac{s}{s + \mu_h} \left[1 - \left(\frac{K_1 \mu_l}{s + \mu_h + K_1 \mu_l} \right)^{j-K_1} \right] \quad (28)$$

3. $B_{ij} \in \{(1, K_2 + 1), \dots, (1, N_l)\}$

In this case the ltype source is blocked while the htype source is transferring a burst, but there are more than K_2 active ltype sources. So although the htype source is served, the ltype source can still remain blocked. Let S_h be the service time of the htype source and $S_l^{(n)}$ the service time of n ltype sources while the htype is being served. Let us consider the density of the blocking time. For convenience of notation we define $P_{ij}(T_l = x) = P(T_l = x | B_{ij})$. Clearly

$$P_{ij}(T_l = x) = P_{ij}(T_l = x, S_h < S_l^{(1)}) +$$

$$\sum_{k=1}^{j-K_2-1} P_{ij}(T_l = x, S_l^{(k)} < S_h < S_l^{(k+1)}) + P_{ij}(T_l = x, S_h > S_l^{(j-K_2)}) \quad (29)$$

After some computation, the Laplace transform of the previous expression yields:

$$F_{T_l|B_{ij}}(s) = \sum_{k=0}^{j-K_2-1} \mu_h \left(\frac{K_2 \mu_l}{s + K_2 \mu_l} \right)^{j-k-K_2} \frac{(K_1 \mu_l)^k}{(s + \mu_h + K_1 \mu_l)^{k+1}} +$$

$$\left[1 - \frac{s}{s + \mu_h} \left[1 - \left(\frac{K_1 \mu_l}{s + \mu_h + K_1 \mu_l} \right)^{K_2-K_1} \right] \right] \left(\frac{K_1 \mu_l}{s + \mu_h + K_1 \mu_l} \right)^{j-K_2} \quad (30)$$

Inversion of 26, 28 and 30, and substitution into 11 yields the distribution of the blocking time T_l . Differentiating these equations we calculate the mean blocking time

$$\bar{T}_l = \sum_{\forall B_{ij}} -\frac{d}{ds} F_{T_l|B_{ij}}(s) \Big|_{s=0} P(B_{ij}) \quad (31)$$

REFERENCES

- Bernstein G. M. and Nguyen D. H. (1994) Blocking Reduction in Fast Reservation Protocols, *IEEE INFOCOM'94*, **9c.2**, 1208–15.
- Boyer P.E. and Tranchier D.P. (1992) A reservation principle with applications to the ATM traffic control. *Computer Networks and ISDN Systems*, **24**, 321–324.
- Enssle J., Briem U. and Kröner H. (1994) Performance Analysis of Fast Reservation Protocols for ATM, *IFIP TC6 2nd Workshop on Performance Modelling and Evaluation of ATM Networks*, pp. 24/1–24/9.
- ITU (1995) *ITU-T Recommendation I.371, Traffic Control and Congestion Control in B-ISDN (Temporary Document)*, ITU Telecommunication Standardization Sector Study Group 13, Geneva.
- Neuts M.F. (1989) Chapter 1: The M/G/1 Queue and Some of its Variants, in *Structured Stochastic Matrices of M/G/1 Type and Their Applications*, Marcel Dekker, New York.
- Suzuki H. and Tobagi F.A. (1992) Fast Bandwidth Reservation Scheme with multi-Link & Multi-Path Routing in ATM Networks, *IEEE INFOCOM'92*, **10A.2**, 2233–40.
- Yin N. and Hluchyj M.G. (1994) On Closed-Loop Rate Control for ATM Cell Relay Networks, *IEEE INFOCOM'94*, **1c.4**, 99–108.

BIOGRAPHY

Llorenç Cerdà graduated in Telecommunications Engineering from Polytechnic University of Catalonia (UPC) in 1993. He joined UPC in 1994 and currently he is Assistant Professor and a PhD student in the Computer Architecture Department.

Jorge García graduated and received his PhD in Telecommunications Engineering from UPC in 1988 and 1992, respectively. He joined UPC in 1988 and currently is Associate Professor of the Computer Architecture department. In 1992-93 he was Visiting Scientist at the Systems and Industrial Department of the University of Arizona with a NATO fellowship. He has been involved in several RACE projects and currently he is involved in COST-242 project.

Olga Casals graduated and received her PhD from UPC in 1983 and 1986 respectively, both in Telecommunications Engineering. She joined UPC in 1983 where she became Full Professor in 1994 and she is head of a research group on traffic in B-ISDN communications systems. She has been working on ATM networks since 1988 with her participation in the RACE project R1022. She has been also involved in the RACE projects EXPLOIT and BAF and currently she is involved in ACTS project EXPERT and in COST-242.

Efficient Simulation of Consecutive Cell Loss in ATM Networks

V.F. Nicola and G.A. Hagesteijn

Tele-Informatics and Open Systems

University of Twente

P.O. Box 217, 7500 AE, Enschede, The Netherlands.

Telephone: 053-4894286. Fax: 053-4893247. e-mail: vfn@cs.utwente.nl

Abstract

In some B-ISDN applications running on ATM networks (e.g., for audio/video connections), the occasional loss of a single ATM cell may not affect the user's perceived QoS requirement. However, the QoS may be degraded due to the loss of a multiple (consecutive) ATM cells. As the event of consecutive cell loss is (typically) rare, its probability cannot be estimated efficiently using standard simulation. In this paper we propose a fast simulation method, based on importance sampling, to efficiently estimate the probability of a rare consecutive-cell-loss event. As an example, we consider a queueing model of the Leaky Bucket source policing algorithm, operating in a bursty traffic environment. We present empirical results to demonstrate the validity and effectiveness of our fast simulation method.

Keywords

Rare event simulation, Importance sampling, Cell loss, ATM networks, Quality of service

1 INTRODUCTION

In an Asynchronous Transfer Mode (ATM) network, data is transported in fixed-size cells. A cell loss may occur due to a variety of reasons, such as buffer overflow in one or more of the network nodes, or as a result of traffic policing at the interface between the user and the network. In any case, the impact of a cell loss on the quality of service (QoS) provided by a given connection depends on the application and its resilience with respect to such a cell loss.

Due to the bursty nature of traffic generated by broadband applications (e.g., multimedia and video conferencing), cells are likely to be lost in multiples (i.e., losing more than one consecutive arriving cells). For example, a buffer overflow at a network node (even if rare) may result in the loss of many consecutive cells. Recovery techniques, such as cell retransmission, may be implemented at the communication protocol level or at the application level. In some applications (such as packet audio/video communication), the occasional loss of one or a few cells may not influence the QoS. Also, extrapolation and/or error correcting techniques can be used to compensate for such cell loss. However, in the absence of cell retransmission or other adequate recovery procedures, the loss of consecutive ATM cells may lead to a remarkable or intolerable degradation of QoS. Therefore, for most applications, it is important to keep the occurrence of consecutive cell loss as rare as possible. This is particularly true for applications with bursty traffic, for which the frequency of consecutive cell loss tend to be (relatively) high. The number of consecutive cell loss that can be tolerated without affecting the QoS depends on the application and/or the supporting recovery (or error correcting) mechanism, if any. For a given application, it is desirable to keep the frequency of losing more than a certain (tolerable) number of consecutive cells below some acceptable threshold. This frequency may be defined as the reciprocal of the steady-state average number of cells between such consecutive-cell-loss events. In a simple queuing model with a finite buffer, this frequency is closely related to another measure of interest; namely, the probability of consecutive cell loss, say, in a busy cycle.

Needless to say, the development of models for the analysis of consecutive cell loss is of much interest for the proper dimensioning of various buffers and other network control parameters. To the best of our knowledge, so far, there has been no analytical results relating to this relevant problem. For a simple $M/M/1$ queue with a finite buffer, we derive analytic closed form expressions for the frequency of consecutive cell loss and the probability of its occurrence in a busy cycle (see Section 2.2 of this paper.) However, for a $GI/GI/1$ queue, the analysis is considerably more difficult, and a useful analytical or algorithmic solution, if at all possible, is not yet available. For the typically correlated and bursty arrival processes, the feasibility of a useful analysis seems even more remote. Furthermore, the probabilities of interest are typically very small, leading to numerical problems.

In order to avoid restrictions necessary for analytic tractability and/or numerical feasibility, simulation is often preferred for the evaluation of realistic models. However, accurate estimation of the frequency of rare events, such as consecutive cell loss, requires observing numerous such events. But, if the frequency of consecutive cell loss is 10^{-9} per cell, then each consecutive-cell-loss event takes place approximately once in 10^9 cells. Observing a sufficiently large number of consecutive-cell-loss events will take extremely long simulation time.

Importance sampling (Hammersley and Handscomb 1964) has been used effectively to achieve significant speed ups in simulations involving rare events, such as failure in a reliable computer system or cell loss in an ATM communication network. See Nicola et al. (1993) for a review of techniques for fast simulation of highly dependable systems, and Heidelberger (1993) for a survey of efficient simulation methods to estimate buffer overflow probabilities in communication systems. The basic idea of importance sampling is to simulate the system under a different probability measure (i.e., with different underlying probability

distributions), so as to increase the probability of typical sample paths involving the rare event of interest. For each sample path (observation) during the simulation, the measure being estimated is multiplied by a correction factor, called the *likelihood ratio*, to obtain an unbiased estimate of the measure in the original system. Asymptotically optimal change of measures (to use in importance sampling) have been found to estimate small probabilities of buffer overflow in relatively simple queueing models (see, Parekh and Walrand (1989), Sadowsky (1991), Chang et al. (1993) and others.) In this paper, we develop heuristics, which are partly based on these optimal change of measures, to estimate very small consecutive-cell-loss probabilities in simple $GI/GI/1/k$ queues (k is the buffer capacity, including the server). We use our heuristics to evaluate a queueing model of the Leaky Bucket (LB) algorithm (see Rathgeb (1991)). Two cell arrival processes are considered; namely, a Poisson process (mainly for validation and experimentation) and a bursty two-phase burst/silence process (see Section 4.4). Empirical results demonstrate the effectiveness of our method to estimate very small consecutive-cell-loss probabilities. These results also show that the simulation time needed to achieve a given accuracy increases (however, slightly) with the number of consecutive cell loss. This increase is attributed to the inherent increase in variability of the probability being estimated, rather than the rarity of the event.

The rest of this paper is organized as follows. In Section 2, we introduce some notation relevant to the study of consecutive cell loss in simple queues, and we carry out the analysis for the $M/M/1/k$ queue. In Section 2.3, we briefly introduce the problem of rare event simulation and review the basic idea of importance sampling. Change of measures used in importance sampling to speed up simulations of simple queues are presented in Section 3; both, a rare full-buffer event and a rare consecutive-cell-loss event, are considered. Validation and experiments with our heuristic change of measure to simulate a queueing model of the LB algorithm are presented in Section 4. Conclusions are given in Section 5.

2 CONSECUTIVE CELL LOSS IN SIMPLE QUEUES

In this section we give brief preliminaries and notation that are needed for the discussion of consecutive cell loss in simple queues. For an $M/M/1/k$ queue, i.e., Poisson cell arrivals and exponential service time distribution, the analysis is not complicated and it is carried out in this section. The results of this analysis are used in Section 4 to validate statistical output obtained from simulation. For general inter-arrival and/or service time distributions, the analysis is considerably more difficult and is not considered here.

2.1 Preliminaries

Consider an $GI/GI/1/k$ queue (k is the buffer capacity, including the server). The probability density function (pdf) of the inter-arrival (resp., service) time is given by $f_A(t)$ (resp., $f_S(t)$.) Define the n -consecutive-cell-loss event to be the (cell arrival) event at which exactly n consecutive cells are lost during a single full-buffer (or overflow) period. (Note that more than n cells may be lost during the same overflow period.) We are interested in the steady-state frequency of this event, i.e., the reciprocal of the average number of arriving cells between two subsequent n -consecutive-cell-loss events; this is denoted by \mathcal{F}_n . A closely related measure of interest is the probability of n or more consecutive cell losses in a busy cycle; this is denoted by γ_n .

Let $N(t)$ be the number of items (cells) in the queue (including that in service) at time t , and denote by $t_j, j = 0, 1, 2, \dots$, the consecutive instants in time at which $N(t)$ jumps from 0 to 1, i.e., for all $j = 0, 1, 2, \dots$,

$N(t_j^-) = 0$ and $N(t_j^+) > 0$. Define a *busy cycle* to be the evolution of the process $N(t)$ between two such consecutive instants, say, t_j and t_{j+1} . Note that $t_j, j = 0, 1, 2, \dots$, constitute renewal points, and, therefore, busy cycles are i.i.d. (independent and identically distributed.) The length of a busy cycle is a r.v. T ; for the j -th busy cycle $T_j = t_j - t_{j-1}, j = 1, 2, \dots$. The number of arrivals during a busy cycle is a r.v. N which, because of buffer overflow, is not necessarily equal to the number of departures in the same busy cycle; for the j -th busy cycle it is denoted by N_j . Furthermore, denote by $O_{n,j}$ the number of full-buffer periods in the j -th cycle during which n or more cells are lost. $O_{n,j}$ is a realization of the random number O_n . It follows that the reciprocal of the long-run (steady-state) average number of arriving cells between two n -consecutive-cell-loss events, i.e., the frequency \mathcal{F}_n , is given by

$$\mathcal{F}_n = \frac{E(O_n)}{E(N)}. \quad (1)$$

Usually, analytic (or numerical) solution for $E(N)$ can be determined. In particular, for an $M/G/1/k$ queue, it is simply given by $1/p_I$, where p_I is the steady-state probability that the server is idle (see, for example, Cooper (1981)). The analysis for $E(O_n)$ is considerably more complicated, mainly because the length of a full-buffer period depends on the sample path (within a busy cycle) leading to that full-buffer. For example, in an $M/G/1/k$ queue, full-buffer periods in the same busy cycle are independent, but the first full-buffer period has a different distribution from that of the second and all subsequent full-buffer periods. However, in an $M/M/1/k$ queue, all full-buffer periods are independent and have the same exponential (service time) distribution, regardless of the sample path leading to the full-buffer. This independence yields significant simplifications leading to the analytical results obtained in the following section.

2.2 Analysis of the $M/M/1/k$ Queue

Consider an $M/M/1/k$ queue with an arrival rate λ and a service rate μ . A busy cycle is defined as above. Define $\pi_i, 0 \leq i \leq k$ as the probability that the number in the system, $N(t)$, moves from level i to level k without hitting level 0. In other words, given that $N(t) = i$, π_i is the probability that the full-buffer state will be reached before the end of the busy cycle. Let γ be the probability of at least one full-buffer period in a busy cycle. Furthermore, given a full-buffer, let ϕ be the probability of yet another full-buffer period in the same busy cycle. It follows that $\gamma = \pi_1$ and $\phi = \pi_{k-1}$. The probabilities $\pi_i, 0 \leq i \leq k$ can be determined from the following equations

$$\pi_i = \frac{\mu}{\lambda + \mu} \pi_{i-1} + \frac{\lambda}{\lambda + \mu} \pi_{i+1}, \quad 1 \leq i \leq k-1, \quad (2)$$

with $\pi_0 = 0$ and $\pi_k = 1$. It follows that

$$\pi_i = \frac{\left(\frac{\mu}{\lambda}\right)^i - 1}{\left(\frac{\mu}{\lambda}\right)^k - 1}, \quad 1 \leq i \leq k-1. \quad (3)$$

Now, let p_n be the probability of n or more arrivals (i.e., n or more consecutive losses) in a single full-buffer period. Since full-buffer periods are independent and having the same exponential distribution with a mean $1/\mu$, it follows that

$$p_n = \left(\frac{\lambda}{\lambda + \mu}\right)^n, \quad 0 \leq n. \quad (4)$$

$P(O_n \geq i)$ is the probability, in a busy cycle, of i or more full-buffer periods, during each of which there are n or more (lost) arrivals. The probability of (at least one) n -consecutive-cell-loss in a busy cycle, γ_n , is given by

$$\begin{aligned} \gamma_n &= P(O_n \geq 1) = \sum_{k=1}^{\infty} \gamma \phi^{k-1} (1 - p_n)^{k-1} p_n \\ &= \frac{\gamma p_n}{1 - \phi(1 - p_n)}. \end{aligned} \quad (5)$$

Also, define ϕ_n to be the probability of another n -consecutive-cell-loss in the same busy cycle. Then

$$\begin{aligned} \phi_n &= \sum_{k=1}^{\infty} \phi^k (1 - p_n)^{k-1} p_n \\ &= \frac{\phi p_n}{1 - \phi(1 - p_n)}. \end{aligned} \quad (6)$$

It follows that

$$P(O_n \geq i) = \gamma_n \phi_n^{i-1}, \quad i \geq 1, \quad (7)$$

and

$$E(O_n) = \frac{\gamma_n}{1 - \phi_n} = \frac{\gamma p_n}{1 - \phi}. \quad (8)$$

Note that for a sufficiently high number of consecutive losses $p_n \ll 1$ and $E(O_n) \approx \gamma_n$.

The above analysis is not valid for other queues, such as $M/G/1/k$ and $GI/M/1/k$. Appropriate analysis techniques may be developed for these queues, which is a subject for further investigation and is not considered in this paper. For these and other $GI/GI/1/k$ queues, we use simulation to estimate $E(O_n)$ and/or γ_n . However, because the n -consecutive-cell-loss is typically a rare event, $E(O_n)$ and γ_n are very small quantities, difficult to estimate using standard simulation. In the next section, we develop fast simulation methods, based on importance sampling, to efficiently estimate γ_n and/or $E(O_n)$. These methods can be validated by comparing statistical output from simulations of the $M/M/1/k$ queue with the above analytical results.

2.3 IMPORTANCE SAMPLING

In a $GI/GI/1/k$ queue, let us consider the estimation of the probability of reaching full-buffer in a busy cycle, γ (see Section 2 for notation). This probability can be expressed as $\gamma = E_f(I(T_{fb} < T))$, where T_{fb} is a r.v. denoting the time to reach a full buffer in a busy cycle, and T is a r.v. denoting the cycle time (as defined in Section 2). $I(\cdot)$ is the indicator function. Note that $T_{fb} = \infty$ for a busy cycle in which the buffer is never full. The subscript f denotes the underlying original probability measure (i.e., the original arrival and service processes). Using standard simulation we generate n independent busy cycles to obtain samples of $I(T_{fb} < T)$, say, I_1, I_2, \dots, I_n . Then $\hat{\gamma} = \sum_{i=1}^n I_i/n$ is an unbiased estimator of γ . The variance of this estimator is given by $Var_f(I(T_{fb} < T))/n$, where $Var_f(I(T_{fb} < T)) = E_f(I^2(T_{fb} < T)) - E_f^2(I(T_{fb} < T)) = \gamma - \gamma^2$. From the central limit theorem (CLT) we have $\sqrt{n}(\hat{\gamma} - \gamma) \rightarrow N(0, Var_f(I(T_{fb} < T)))$. The CLT approximation can be used to obtain a 99% confidence interval (CI), the half width (HW) of which is given by $2.56 \sqrt{Var_f(I(T_{fb} < T))/n}$. The relative error (RE) is defined as the ratio $HW/\gamma \approx 2.56/\sqrt{n\gamma}$. Obviously, for a fixed n , $RE \rightarrow \infty$ as $\gamma \rightarrow 0$. This is the problem when using standard simulation to estimate the probability of a rare event, such as γ . Importance sampling can be used to overcome this inherent problem.

Now, let g be another underlying probability measure, and ω be a sample path (e.g., a busy cycle) in the set Ω of all possible sample paths. Denote by $dg(\omega)$ the probability of the sample path ω according to the new probability measure g . (Similarly, $df(\omega)$ is the probability of the sample path ω according to the original probability measure f .) Note that γ can be written as follows

$$\begin{aligned} \gamma &= \int_{\omega \in \Omega} I_{\omega}(T_{fb} < T) df(\omega) = \int_{\omega \in \Omega} I_{\omega}(T_{fb} < T) \frac{df(\omega)}{dg(\omega)} dg(\omega) \\ &= \int_{\omega \in \Omega} I_{\omega}(T_{fb} < T) L(\omega) dg(\omega) = E_g(I(T_{fb} < T)L), \end{aligned} \quad (9)$$

where $I_{\omega}(\cdot)$ is the indicator function evaluated for sample path ω , and $L(\omega) = df(\omega)/dg(\omega)$ is the likelihood ratio. It is clear from the above equation that the only condition imposed on the new probability measure g is: $dg(\omega) > 0$ whenever $I_{\omega}(T_{fb} < T) df(\omega) > 0$. It follows that we can simulate the system using the new probability measure g to obtain n independent samples of $I(T_{fb} < T)L$, say, $I_1L_1, I_2L_2, \dots, I_nL_n$. An unbiased estimate of γ is given by $\hat{\gamma} = \sum_{i=1}^n I_iL_i/n$. The variance of this estimator is $Var_g(I(T_{fb} < T)L)/n = (E_g(I(T_{fb} < T)L^2) - \gamma^2)/n$. Notice that a zero variance estimator is obtained if we choose the new probability measure g such that for all $\omega \in \Omega$, $dg(\omega) = I_{\omega}(T_{fb} < T) df(\omega)/\gamma$. However, this is not possible, since it requires the knowledge of γ , the quantity we are trying to estimate! The main challenge in importance sampling is to find a robust and easily implementable new probability measure g such that

$$E_g(I(T_{fb} < T)L^2) = E_f(I(T_{fb} < T)L) \ll E_f(I(T_{fb} < T)). \quad (10)$$

This means that the variance of the importance sampling estimate is much less than the variance of the standard simulation estimate. In other words, for the same simulation effort (e.g., the same number of busy cycles n), importance sampling yields an estimate with much smaller relative error than that obtained using standard simulation. (This also implies a significant speed up of simulation time to achieve certain accuracy.) Notice from the above equation that much variance reduction is obtained if $L(\omega) = df(\omega)/dg(\omega) \ll 1$ whenever $I_{\omega}(T_{fb} < T) = 1$. That is, g should be chosen so as to significantly increase the probability of the rare event $\{T_{fb} < T\}$. An "effective" change of probability measure, g , is one for which the relative error (RE) remains bounded, also as the probability of the rare event tends to zero. This is

a desirable property which implies that the simulation effort (e.g., the number of samples n) to achieve a given relative error remains the same as the rare event becomes rarer. In some cases, this property may be established empirically for a given importance sampling technique, as will be demonstrated in our experimental results of Section 4.

3 FAST SIMULATION OF SIMPLE QUEUES

Consider a simple queue with a finite buffer. The cell arrival “rate” is assumed to be sufficiently smaller than the service “rate”, so that reaching a *full-buffer* (or buffer overflow) is a rare event. Efficient simulation involving a rare full-buffer event has been considered by many (see, for example, Parekh and Walrand (1989) and Sadowsky (1991).) Another rare event of interest is the *n-consecutive-cell-loss* event, which may occur only after the full-buffer is reached. In this section we consider these two related rare events, and develop an importance sampling heuristic to speed up simulations involving a rare consecutive-cell-loss event.

3.1 Rare Full-Buffer Event

In a $GI/GI/1/k$ queue, let us again consider the estimation of the probability of reaching full-buffer in a busy cycle, γ . As in Section 2.3, this probability can be expressed as $\gamma = E_f(I(T_{fb} < T))$, where the expectation is taken with respect to the original probability measure f . Since $\{T_{fb} < T\}$ is a rare event (i.e., $\gamma \approx 0$), using standard simulation is very inefficient, as it yields 0 for the indicator function on almost all busy cycles. Using importance sampling, we have $\gamma = E_f(I) = E_g(IL)$, where f and g are the original and the new probability measures, respectively, and L is the likelihood ratio. Denote by $dg(\omega)$ the probability of a sample path ω according to the new probability measure g . (Similarly, $df(\omega)$ is the probability of a sample path ω according to the original probability measure f .) Then $L(\omega) = df(\omega)/dg(\omega)$ is the likelihood ratio associated with a sample path ω ; it can be computed easily during the simulation. For example, let $t_{A,j}^i$ (resp., $t_{S,j}^i$), $i = 1, 2, \dots, N_j$, be the cell arrival (resp., departure) instants in the j -th busy cycle. Furthermore, let $g_{A,j}^i(t)$ (resp., $g_{S,j}^i(t)$) be the new i -th inter-arrival (resp., service) time density used to simulate the system with importance sampling. The likelihood ratio, L_j , associated with the j -th busy cycle, takes the form

$$L_j = \prod_{i=1}^{N_j} \frac{f_{\Lambda}(t_{A,j}^{i+1} - t_{A,j}^i)}{g_{\Lambda,j}^i(t_{A,j}^{i+1} - t_{A,j}^i)} \times \frac{f_S(t_{S,j}^i - t_{S,j}^{i-1})}{g_{S,j}^i(t_{S,j}^i - t_{S,j}^{i-1})}. \quad (11)$$

Note that $t_{S,j+1}^0 = t_{\Lambda,j}^{N_j+1} = t_{\Lambda,j+1}^1$ is the instant at which the j -th busy cycle ends and the $j+1$ -th busy cycle begins. Thus, L_j can be computed recursively at arrival and departure events during the simulation.

Now, let b be the number of independent “biased” (using importance sampling) busy cycles used to obtain estimates for the mean and the variance of the r.v. IL . These estimates are given by

$$\hat{\mu}_I = \sum_{j=1}^b I_j L_j / b, \quad \hat{\sigma}_I^2 = \sum_{j=1}^b (I_j L_j - \hat{\mu}_I)^2 / (b-1).$$

From the central limit theorem, for large b , the estimate $\hat{\mu}_I$ is approximately normally distributed. It follows that the relative half-width (in percentage) of the 99% confidence interval for the above estimator is given by $2.56(\hat{\sigma}_I/\hat{\mu}_I) \times 100$.

In the following we consider the optimal change of measure (importance sampling distribution) to efficiently estimate γ . Let $F_A(\theta) = \int_{t=0}^{\infty} e^{\theta t} f_A(t) dt$ be the moment generating function of the inter-arrival times. Define $f_A^\theta(t) = e^{\theta t} f_A(t)/F_A(\theta)$; this is another pdf obtained by exponentially tilting (twisting) the pdf $f_A(t)$ at a parameter θ . Similarly, $F_S(\theta) = \int_{t=0}^{\infty} e^{\theta t} f_S(t) dt$ is the moment generating function of the service times, and $f_S^\theta(t) = e^{\theta t} f_S(t)/F_S(\theta)$ is the corresponding exponentially tilted pdf.

Using heuristic arguments based on the theory of large deviations (Bucklew 1990), Parekh and Walrand (1989) proposed an importance sampling distribution to efficiently estimate the probability of buffer overflow in a $GI/GI/1/k$ queue. In Sadowsky (1991), this distribution was proved to be the unique asymptotically (as $k \rightarrow \infty$) optimal change of measure. Let θ^* be the solution of the equation

$$F_A(-\theta^*) F_S(\theta^*) = 1. \quad (12)$$

Then the optimal change of measure is obtained by simulating the $GI/GI/1/k$ queue with the exponentially tilted densities $g_A(t) = f_A^{-\theta^*}(t)$ and $g_S(t) = f_S^{\theta^*}(t)$. Importance sampling is "turned on" at the start of each busy cycle, and is "turned off" at the occurrence of the rare event. The moment generating functions for the new (optimal) inter-arrival and service times are given by

$$G_A(\theta) = \frac{F_A(\theta - \theta^*)}{F_A(-\theta^*)}, \quad G_S(\theta) = \frac{F_S(\theta + \theta^*)}{F_S(\theta^*)}. \quad (13)$$

Consider the $M/M/1/k$ queue with its arrival rate λ much smaller than its service rate μ (i.e., $\lambda \ll \mu$), so that a full buffer is a rare event. $F_A(-\theta) = \lambda/(\lambda + \theta)$ and $F_S(\theta) = \mu/(\mu - \theta)$, for $\theta < \mu$. Solving the equation $F_A(-\theta^*) F_S(\theta^*) = 1$ for θ^* , we get $\theta^* = \mu - \lambda$. It follows that $G_A(\theta) = \mu/(\mu - \theta)$ and $G_S(\theta) = \lambda/(\lambda - \theta)$, i.e., optimally, the $M/M/1/k$ queue is simulated with arrival rate μ and service rate λ . This change of measure accelerates the arrival process relative to the service process, thus increasing the probability of a full buffer in the simulated system.

In the next section, we use the optimal importance sampling distribution (as outlined above) in a heuristic to estimate very small consecutive-cell-loss probabilities.

3.2 Rare Consecutive-Cell-Loss Event

In this section we consider the estimation of the probability of losing n or more consecutive cells in a busy cycle, γ_n (see Section 2 for notation). This probability can be expressed as $\gamma_n = E_f(I(T_n < T))$, where the expectation is taken with respect to the original probability measure f . T_n is a r.v. denoting the time to the first n -consecutive-cell-loss event in a busy cycle, and T is a r.v. denoting the cycle time (also defined in Section 2). Note that $T_n = \infty$ for a busy cycle in which there is no n consecutive cell loss. Here too, since $\{T_n < T\}$ is a rare event, using standard simulation is very inefficient. In fact, the event $\{T_n < T\}$ must be at least as rare as the event $\{T_{fb} < T\}$, since the former may or may not occur only after the latter has occurred. Using importance sampling, we have $\gamma_n = E_f(I) = E_g(IL)$, where f and g are the original and the new probability measures, respectively, and L is the likelihood ratio. Based on b independent "biased" (using importance sampling) busy cycles, estimates of the mean $\hat{\mu}_I$ and the variance $\hat{\sigma}_I^2$ (and hence confidence intervals) are obtained as described in Section 3.1.

To the best of our knowledge, the problem of estimating the probability of a rare consecutive-cell-loss event (γ_n) using importance sampling has not been considered before. Note that this rare event can only occur during a full-buffer period, i.e., after the occurrence of a typically rare full-buffer event. Therefore, it seems intuitive to use two “biasing” (importance sampling) schemes, one to reach a full-buffer, and another, if necessary, to lose n consecutive cells during that full-buffer period. The main idea of our importance sampling heuristic is to use the optimal change of measure to reach the full-buffer state (as described in Section 3.1.) Once (and every time, until the consecutive loss of n cells) the full-buffer state is reached, additional “biasing” (e.g., by increasing the arrival “rate”) is applied (if necessary) to increase the probability of n or more arrivals (losses) during the full-buffer period. “Biasing” is turned off as soon as the rare event of interest occurs, i.e., n arrivals during a full-buffer period. Otherwise, “biasing” is continued according to the optimal change of measure (of Section 3.1) until the next full-buffer period or the end of the busy cycle. The implementation details of “biasing” during full-buffer periods may differ depending on the particular arrival and service processes being considered. These details will be discussed for each of the models used in our experiments of Section 4. Empirical results from these experiments demonstrates the effectiveness of the above importance sampling heuristic to estimate γ_n . The same heuristic can also be used to estimate \mathcal{F}_n , the frequency of the n -consecutive-cell-loss event. In either case, several orders of magnitude “speed ups” over standard simulation can be obtained.

It is important to mention that, in general, the simulation effort (with importance sampling) slowly increases with the number of consecutive cell loss of interest, i.e., the importance sampling scheme is not asymptotically (as $n \rightarrow \infty$) efficient. (This can, perhaps, be seen from the experimental results for the $M/D/1/k$ queue in Section 4.3.) However, this is not due to the increased rarity of the n -consecutive-cell-loss event, but due to increase in the inherent variance of the probability of n or more arrivals during a full-buffer period. Let V be a r.v. denoting the length of a full-buffer period, then for Poisson arrivals with a rate λ , this probability is given by $P_n(V) = e^{-\lambda V} \sum_{i=n}^{\infty} (\lambda V)^i / i!$. Clearly, the variance of $P_n(V)$ increases with the variance of V and is amplified for high values of n . It is this inherent increase in variability which cannot be reduced by importance sampling. In fact, for an $M/M/1/k$ queue, the full-buffer periods, V , are independent and exponentially distributed with a mean $1/\mu$. In this case, samples of $P_n(V)$ observed during simulation can be replaced by their (deterministic) mean $p_n = (\frac{\lambda}{\lambda+\mu})^n$. This way, the variability of $P_n(V)$ does not affect the simulation results. Indeed, for an $M/M/1/k$ queue, this special implementation of our heuristic is asymptotically efficient (as $n \rightarrow \infty$), which is clearly demonstrated by the empirical results in Section 4.1.

4 EXPERIMENTAL RESULTS

In this section we use fast simulation methods discussed in Sections 3.1 and 3.2 to evaluate a model of the Leaky Bucket (LB) algorithm. For validation purposes, the simulation of an $M/M/1/k$ queue is considered in Section 4.1. The operation of the LB algorithm and its model are described in Section 4.2. The evaluation of this model is considered in Sections 4.3 and 4.4, for Poisson and two-phase burst/silence (TPBS) cell arrival processes, respectively. The empirical results displayed here include estimates of γ_n (i.e., the probability of losing n or more consecutive cells in a busy cycle), $E(O_n)$ (i.e., the expected number of n -consecutive-cell-loss events in a busy cycle) and \mathcal{F}_n (i.e., the steady-state frequency of the n -consecutive-cell-loss event.)

4.1 Simulation of the $M/M/1/k$ Queue

In this section we consider the efficient simulation of an $M/M/1/k$ queue to estimate the probability of consecutive cell loss in a busy cycle. For this model, analytical results in Section 2.2 can be used to validate statistical output from simulation. As outlined in Section 3.2 our importance sampling heuristic makes use of two different “biasing” schemes. The first is optimal “biasing” (as described in Section 3.1) to reach the full-buffer state (i.e., the $M/M/1/k$ queue is simulated with arrival rate μ and service rate λ .) The second is “biasing” during full-buffer periods, which in the special case of an $M/M/1/k$ queue can be implemented as follows. As argued in Section 3.2, the probability of n or more arrivals (losses) during a full-buffer period is given by $p_n = (\frac{\lambda}{\lambda+\mu})^n$, which is typically very small in the original queue. In the simulated queue, we increase this probability to p_s (a constant sufficiently higher than p_n ; for example, $p_s = 0.5$). With probability p_s , the full-buffer period is considered to be a “successful” overload period (i.e., having n or more arrivals). Let U be a uniform random variable ($0 < U < 1$). Every time (until the consecutive loss of n cells) the full-buffer state is reached, we take a sample u of U . If $u \leq p_s$, then the n -consecutive-cell-loss event is considered to have occurred, and “biasing” is turned off until the end of the current busy cycle. In this case, the likelihood ratio is updated by the multiplication factor p_n/p_s . (Note that in this implementation, a sample of the full-buffer period need not be generated, and the simulation is continued, from the instant of reaching the full-buffer state, as if a departure event has just occurred leaving the queue with $k - 1$ cells.) Otherwise, if $u > p_s$, then the n -consecutive-cell-loss event is considered to have not occurred, and “biasing” is continued as described in Section 3.1 until the next full-buffer period or the end of the current busy cycle. In this case, the likelihood ratio is updated by the multiplication factor $(1 - p_n)/(1 - p_s)$.

Now let us consider the $M/M/1/k$ queue with $\lambda = 0.8$ cells per unit of time, $\mu = 1.0$ cells per unit of time and $k = 25$. In Table 1, for increasing n , we give fast simulation estimates of the cycle-based quantities; namely, the n -consecutive-cell-loss probability (γ_n) and the expected number of n -consecutive-cell-loss events $E(O_n)$. Numerical results from analysis are also displayed. Consistent with our remark in Section 2.2, note that $E(O_n) \approx \gamma_n$ for values of $n \geq 8$. Also, Note that the frequency \mathcal{F}_n can be determined by $E(O_n)/E(N) = P_l E(O_n)$, where $P_l = 1 - \frac{\lambda}{\mu}$.

Using different arrival and service rates, experiments indicate that for high n , the lowest relative error can be obtained by setting p_s (approximately) to $1 - \frac{\lambda}{\mu}$. Therefore, the “biasing” probability p_s is heuristically set to $\max(p'_n, 1 - \frac{\lambda}{\mu})$, where p'_n is the (new) probability of n or more arrivals during a full-buffer period in the simulated system (i.e., with the optimal change of measure as given in Section 3.1.) For the simulated $M/M/1/k$ queue, it follows that $p'_n = (\frac{\mu}{\lambda+\mu})^n$. 25600 “biased” busy cycles were simulated to get the estimates and their relative error (i.e., the relative half-width of the 99% confidence interval) in percentage. Note that fast simulation results are in good agreement with the numerical results from analysis. Also, the relative error does not increase for larger values of n ; this verifies the asymptotic optimality of the particular implementation of our proposed importance sampling method when applied to the $M/M/1/k$ queue.

4.2 The Leaky Bucket (LB) Algorithm

An ATM connection is established with an admission contract which specifies the traffic characteristics of the source and the quality of service (QoS) to be guaranteed by the network. In order for the network to ensure that the admission contract is not violated, the usage parameter control (UPC) procedure is invoked to monitor the actual traffic and to police the excess traffic violating the contract. The Leaky

Bucket (LB) algorithm is a popular UPC procedure and can easily be implemented with counters (see Turner (1986).) Each time a cell arrives, the counter is incremented by one. As long as the counter has a positive value, it is decremented at fixed intervals, d . When the cell arrival “rate” exceeds the periodic decrement “rate,” the counter value will increase. If the counter reaches a pre-specified limit, say, k , then the source is considered to have exceeded its admission contract, and subsequent cells are discarded (or marked for policing) until the counter value falls below the limit again. The operation of this LB algorithm can be modeled as a $GI/D/1/k$ queue, in which the service time is deterministic and identical to the decrement interval, d . An arriving cell is lost if it finds a full buffer.

For a two-phase burst/silence source model (see Section 4.4), the stationary cell loss probability can be obtained by a numerical method whose complexity grows in proportion to the value of k (Rathgeb 1991.) No analytical or numerical method is available yet to obtain the probability of consecutive cell loss in a $GI/D/1/k$ queue. In order to avoid restrictions necessary for analytic tractability and/or numerical feasibility, simulation is often preferred for the evaluation of realistic models of the LB algorithm. However, standard simulation is not efficient because consecutive cell loss is a rare event. Accurate and efficient estimation of very small probabilities, such as γ , using importance sampling has been considered in Nicola et al. (1994). In the next two sections, we use the importance sampling heuristic proposed in Section 3 to efficiently estimate γ_n , $E(O_n)$ and \mathcal{F}_n in a model of the LB algorithm with (non-bursty) Poisson and (bursty) TPBS cell arrival processes.

4.3 Poisson Cell Arrival Process

In this section we use importance sampling to efficiently estimate the probability of consecutive cell loss in a busy cycle of an $M/D/1/k$ queueing model of the LB algorithm (i.e., for a Poisson cell arrival process). The arrival rate is λ and the service time is a constant d . As outlined in Section 3.1, the optimal change of measure to reach the full-buffer state can be obtained by solving Equation (12) for θ^* . The corresponding inter-arrival and service time densities can now be determined from their generating functions as given in Equation (13). It follows that the optimal service times are also deterministic and identical to the original (i.e., no change in the service process.) However, the arrival process does change, so as to increase the probability of the rare full-buffer event. We note that full-buffer periods (i.e., the actual remaining service time upon reaching the full-buffer state) in the same busy cycle are neither independent nor identically distributed. Therefore, in this implementation, these full-buffer periods must be simulated (unlike the implementation for the $M/M/1/k$ queue). The probability of n or more arrivals (losses) during a full-buffer period depends on the remaining service time ($r < d$) and is given by $P_n(r) = e^{-\lambda r} \sum_{i=n}^{\infty} (\lambda r)^i / i!$. This probability is typically very small in the original system, and, therefore, “biasing” is necessary to increase the probability of “success” (i.e., n or more arrivals) during the full-buffer period. In the simulated queue, we increase this probability to p_s (a constant sufficiently higher than $P_n(r)$; for example, $p_s = 0.5$). Every time (until the consecutive loss of n cells) the full-buffer state is reached, we take a sample u of a uniform random variable U (defined in Section 4.1). If $u \leq p_s$, then the n -consecutive-cell-loss event is considered to have occurred, and “biasing” is turned off until the end of the current busy cycle. In this case, at the end of the full-buffer period, the likelihood ratio is updated by the multiplication factor $P_n(r)/p_s$. Otherwise, if $u > p_s$, then the n -consecutive-cell-loss event is considered to have not occurred, and “biasing” is continued immediately after the full-buffer period (as described in Section 3.1) and until the next full-buffer period or the end of the current busy cycle. In this case, at the end of the full-buffer period, the likelihood ratio is updated by the multiplication factor $(1 - P_n(r))/(1 - p_s)$.

Note that when the full-buffer period r is very small (i.e., $r \ll d$), “biasing” may yield non-typical sample paths, resulting in extremely small values for the likelihood ratio and leading to unstable estimates.

To overcome this problem, the above heuristic is modified as follows. Upon reaching the full-buffer state, $P_n(r)$ is determined, and “biasing” during the full-buffer period (as outlined above) is activated only if, say, $P_n(r)/P_n(d) \geq 4 \times 10^{-3}$. In this way, “biasing” is activated only when a full-buffer period is sufficiently large to yield a rare (but typical) sample path. As long as the consecutive-cell-loss event did not occur, “biasing” to reach the next full-buffer period is resumed as outlined above. The following example shows that the above heuristic with this modification is quite robust and effective.

Now let us consider the model of the LB algorithm with a Poisson cell arrival process at rate $\lambda = 0.8$ cells per unit of time. The new (optimal) arrival process to reach the full-buffer state is also Poisson, however, at an increased rate $\lambda^* = \lambda + \theta^*$, where (from Equation (12)) θ^* is the non-trivial solution of $\lambda + \theta^* = \lambda e^{d\theta^*}$. The (deterministic) service time is set to $d = 1$ time unit, $k = 10$, and we vary the number of consecutive cell loss, n . In Table 2, we list fast simulation estimates of γ_n and $E(O_n)$ as well as their relative error (i.e., the relative half-width of the 99% confidence interval) in percentage. 25600 “biased” busy cycles were used to get these estimates. Using different arrival rates and/or service times, the best relative error (for high values of n) is obtained by setting p_s (approximately) to $1 - \lambda d$. Therefore, the “biasing” probability p_s is heuristically set to $\max(P'_n(r), 1 - \lambda d)$, where $P'_n(r)$ is the (new) probability of n or more arrivals during the full-buffer period r in the simulated system (i.e., with the increased optimal arrival rate λ^*). For the simulated $M/D/1/k$ queue, it follows that $P'_n(r) = e^{-\lambda^* r} \sum_{i=n}^{\infty} (\lambda^* r)^i / i!$. Note that if “biasing” is not activated in a full-buffer period because $r \ll d$, then $p_s = P_n(r)$, and the likelihood ratio is not updated at the end of the full-buffer period. Using the same effort (in CPU time), standard simulation yields meaningful results for only two entries with relatively high probabilities. As can be seen, the relative error of the fast simulation estimates slowly increases with n , which is an indication that the importance sampling heuristic is not asymptotically efficient with respect to n . As explained in Section 3.2, this is due to the increased variability of $P_n(V)$ for higher n , where V is a r.v. denoting the length of a full-buffer period. Note that $E(O_n) \approx \gamma_n$ for values of $n \geq 4$, which validates our remark in Section 2.2 for queues other than the $M/M/1/k$.

4.4 Bursty Cell Arrival Process

In this section we consider the evaluation of the LB algorithm for a more realistic two-phase burst/silence cell arrival process (see Rathgeb (1991)), which we will refer to as TPBS process. This arrival process has been used to model bursty sources, such as packetized voice (see Heffes and Lucantoni (1986)) and interactive data services, and, therefore, it is often used to compare various policing mechanisms. The number of cells per burst is geometrically distributed with a parameter α , and the inter-cell time during a burst is deterministic given by τ . Therefore, transitions from burst to silence occur with a probability α , only at multiples of τ . The duration of the silence phase is exponentially distributed with a mean β^{-1} . The peak cell arrival “rate” is $1/\tau$, and the average cell arrival “rate” $\lambda = (\tau + \alpha/\beta)^{-1}$. Note that we can increase the burstiness of the cell arrival process by increasing the average burst length (i.e., smaller α) while keeping the average cell “rate” the same (i.e., constant α/β .) The pdf of the TPBS inter-arrival time and its moment generating function are given by

$$f_A(t) = \begin{cases} 0, & \text{if } t < \tau, \\ 1 - \alpha, & \text{if } t = \tau, \\ \alpha \beta e^{-\beta(t-\tau)}, & \text{if } t > \tau, \end{cases} \quad (14)$$

$$F_A(\theta) = e^{\theta\tau} \left[(1 - \alpha) + \alpha \frac{\beta}{\beta - \theta} \right]. \quad (15)$$

$g_A(t) = f_A^{-\theta^*}(t)$ is the corresponding exponentially tilted pdf (with a tilting parameter θ^*); its moment generating function is given by $G_A(\theta) = F_A(\theta - \theta^*)/F_A(-\theta^*)$. It can be shown that the tilted pdf, $g_A(t)$, is also a TPBS process with the same deterministic burst inter-cell time τ , and with its parameters, $\beta^* = \beta + \theta^*$, and $\alpha^* = \alpha\beta/(\beta + (1 - \alpha)\theta^*)$. The tilted pdf, $g_A(t)$, is used as the (new) inter-arrival time density for simulation with importance sampling to reach the full-buffer state.

For a TPBS cell arrival process, the LB algorithm can be modelled as $TPBS/D/1/k$ queue. Since the full-buffer and the consecutive cell loss are typically rare events, importance sampling is used to efficiently simulate this system. At the beginning of each busy cycle, and after each full-buffer period (as long as the rare consecutive-cell-loss event has not occurred), “biasing” to reach the next full-buffer period is affected as described in Section 3.1. The new “biased” (TPBS) cell arrival process is determined by α^* , β^* and τ , as given above. The service time is deterministic (d), and, therefore, remains unchanged in the simulated system. As soon as the full-buffer state is reached, further “biasing” during the full-buffer period may be necessary to accelerate the n -consecutive-cell-loss event. Since the inter-cell time during a burst (τ) is deterministic, the number of cells that may be lost during a full-buffer period of length r cannot exceed a maximum given by $n_{max} = \lfloor r/\tau \rfloor$. At the beginning of a full-buffer period of length r , if $n \leq n_{max}$, then “biasing” is done by setting the new β to β^* . If α^* is not sufficient to increase the probability of n or more remaining cells in the current burst to a high value, p_s (for example, $p_s = 0.5$), then the new α is set to α_s as determined from $(1 - \alpha_s)^n = p_s$ (i.e., $\alpha_s = 1 - e^{\ln(p_s)/n}$). In other words, until the consecutive loss of n cells, we use the optimal “biasing” to reach the full-buffer state (i.e., the new α is set to α^* and the new β is set to β^* .) In addition, depending on n and r , more (stronger) “biasing” during the full-buffer period may be necessary (i.e., if $n \leq n_{max}$, then the new α is set to $\min(\alpha^*, \alpha_s)$.) The effectiveness of this heuristic is demonstrated in one example. In another example, we use the heuristic to experiment with the burstiness of the cell arrival process.

In the first experiment, we consider a TPBS cell arrival process with $\alpha = 0.2$, $\beta = 5.0 \times 10^{-4}$ and $\tau = 1$. The (deterministic) service time, d , is set to 100 time units, and k is set to 30. In Table 3, the number of consecutive cell loss, n , is varied, and we give fast simulation estimates of γ_n and \mathcal{F}_n , with their percentage relative error (i.e., the relative half-width of the 99% confidence interval.) 25600 “biased” busy cycles were used to get these estimates. For all n , the “biasing” probability, p_s , is set to 0.5. It is not directly seen from the table, however, it is interesting to point out that, for smaller values of n , stronger “biasing” during full-buffer periods is not necessary (i.e., the new α is set to α^* .) For relatively high consecutive-cell-loss probabilities, it was possible to compare with results from standard simulation using the same effort (in CPU time.) Note that the relative error of the fast simulation estimates slowly increases with n , i.e., the importance sampling heuristic is not asymptotically efficient with respect to n . A similar observation was made in the experiment for the $M/D/1/k$ queue in Section 4.3.

In the second experiment, we consider a TPBS arrival process, in which we increase the burstiness, while fixing the average cell arrival “rate.” As described earlier in this section, this can be achieved by decreasing α and β , while fixing α/β . We set $\tau = 1$ and $\lambda = 1/50$. It follows that α/β is fixed at 49. The (deterministic) service time, d , is set to 25 time units, and k is set to 100. For a fixed number of consecutive cell loss, $n = 5$, in Table 4 we vary the burstiness and give the fast simulation estimates of γ_n and \mathcal{F}_n , with their percentage relative error. 25600 “biased” busy cycles were used to get these estimates. For all values of α , the “biasing” probability, p_s , is set to 0.5. Using the same effort (in CPU time), only for relatively high probabilities, it is possible to obtain meaningful results from standard simulation. As expected, the empirical results in Table 4 indicate a sharp increase in the consecutive-cell-loss probability due to increased burstiness.

5 CONCLUSIONS

In this paper we have proposed a heuristic importance sampling change of measure to efficiently estimate the probability of a rare consecutive-cell-loss event in a $GI/GI/1/k$ queue. This heuristic makes use of the optimal change of measure proposed by Parekh and Walrand (1989) to accelerate the occurrence of a rare full-buffer event in an asymptotically stable queue. However, further “biasing” is necessary to increase the probability of a rare consecutive-cell-loss event during a full-buffer period. Experimental results demonstrate the validity and effectiveness of our fast simulation method, which is used for the evaluation of a $GI/D/1/k$ queueing model of the Leaky Bucket algorithm.

ACKNOWLEDGEMENT

The authors wish to thank Fokke Hocksema for bringing this problem to their attention, and Erik van Doorn for useful discussions on the analysis.

REFERENCES

- Bucklew, J.A. (1990) *Large Deviation Techniques in Decision, Simulation and Estimation*. New York, NY: J. Wiley & Sons, Inc.
- Chang, C.S., P. Heidelberger, S. Juneja and P. Shahabuddin. (1993) Effective bandwidth and fast simulation of ATM intree networks. In *Proc. of the Performance '93 conference*.
- Cooper, R.B. (1981) *Introduction to Queueing Theory*. London: Arnold.
- Hammersley, J.M. and D.C. Handscomb. (1964) *Monte Carlo Methods*. London: Methuen.
- Heffes, H. and D.M. Lucantoni. (1986) A Markov modulated characterization of packetized voice and data traffic and related statistical multiplexer performance. *IEEE J. Select. Areas Commun.* **4**, **6**: 856-868.
- Heidelberger, P. (1993) Fast simulation of rare events in queueing and reliability models. In *Models and Techniques for Performance Evaluation of Computer and Communications Systems*, Springer-Verlag, Lecture Notes in Comp. Sc., **729**: 165-202.
- Nicola, V.F., P. Shahabuddin and P. Heidelberger. (1993) Techniques for fast simulation of highly dependable systems. In *Proc. of the Second International Workshop on Performability Modelling of Computer and Communication Systems*.
- Nicola, V.F., G.A. Hagesteijn and B.G. Kim. (1994) Fast simulation of the Leaky Bucket algorithm. In *Proceedings of the 1994 Winter Simulation Conference*, IEEE Press, 266-273.
- Parekh, S. and J. Walrand. (1989) A quick simulation method for excessive backlogs in networks of queues. *IEEE Trans. Autom. Contr.* **34**, **1**: 54-66.
- Rathgeb, E.P. (1991) Modeling and performance comparison of policing mechanisms for ATM networks. *IEEE J. Select. Areas Commun.* **9**, **3**: 325-334.
- Sadowsky, J.S. (1991) Large deviations theory and efficient simulation of excessive backlogs in a $GI/GI/m$ queue. *IEEE Trans. Autom. Contr.* **36**, **12**: 1383-1394.
- Turner, J.S. (1986) New directions in communications (or which way to the information age?). *IEEE Commun. Mag.* **25**, **10**: 8-15.

Table 1 Estimates of γ_n and $E(O_n)$ in an $M/M/1/k$ Queue

	γ_n		$E(O_n)$	
	Fast Sim.	Anal.	Fast Sim.	Anal.
full-buffer	9.45×10^{-4} $\pm 3.20\%$	9.48×10^{-4}	4.66×10^{-3} $\pm 4.52\%$	4.72×10^{-3}
$n = 1$	7.69×10^{-4} $\pm 3.16\%$	7.58×10^{-4}	2.10×10^{-3} $\pm 4.20\%$	2.10×10^{-3}
$n = 4$	1.58×10^{-4} $\pm 3.25\%$	1.59×10^{-4}	1.81×10^{-4} $\pm 3.50\%$	1.84×10^{-4}
$n = 8$	7.12×10^{-6} $\pm 3.21\%$	7.15×10^{-6}	7.15×10^{-6} $\pm 3.21\%$	7.19×10^{-6}
$n = 16$	1.09×10^{-8} $\pm 3.21\%$	1.09×10^{-8}	1.09×10^{-8} $\pm 3.21\%$	1.09×10^{-8}
$n = 32$	2.53×10^{-14} $\pm 3.21\%$	2.54×10^{-14}	2.53×10^{-14} $\pm 3.21\%$	2.54×10^{-14}
$n = 64$	1.36×10^{-25} $\pm 3.21\%$	1.36×10^{-25}	1.36×10^{-25} $\pm 3.21\%$	1.36×10^{-25}

Table 2 Estimates of γ_n and $E(O_n)$ in an $M/D/1/k$ Queue

	γ_n		$E(O_n)$	
	Std. Sim.	Fast Sim.	Std. Sim.	Fast Sim.
full-buffer	1.00×10^{-2} $\pm 4.49\%$	9.92×10^{-3} $\pm 2.15\%$	4.87×10^{-2} $\pm 5.99\%$	4.80×10^{-2} $\pm 3.21\%$
$n = 1$	6.47×10^{-3} $\pm 4.76\%$	6.38×10^{-3} $\pm 2.22\%$	1.43×10^{-2} $\pm 5.91\%$	1.40×10^{-2} $\pm 3.00\%$
$n = 4$		8.20×10^{-5} $\pm 3.48\%$		8.28×10^{-5} $\pm 3.50\%$
$n = 8$		9.68×10^{-9} $\pm 4.23\%$		9.68×10^{-9} $\pm 4.23\%$
$n = 12$	—	2.15×10^{-13} $\pm 4.97\%$	—	2.15×10^{-13} $\pm 4.97\%$
$n = 16$	—	1.49×10^{-18} $\pm 5.56\%$	—	1.49×10^{-18} $\pm 5.56\%$

Table 3 Estimates of γ_n and \mathcal{F}_n in a *TPBS/D/1/k* Queue

	γ_n		\mathcal{F}_n	
	Std. Sim.	Fast Sim.	Std. Sim.	Fast Sim.
full- buffer	6.14×10^{-3} $\pm 8.46\%$	6.15×10^{-3} $\pm 0.87\%$	1.29×10^{-3} $\pm 9.50\%$	1.28×10^{-3} $\pm 1.86\%$
$n = 1$	5.14×10^{-3} $\pm 9.16\%$	5.19×10^{-3} $\pm 0.87\%$	1.01×10^{-3} $\pm 10.13\%$	1.02×10^{-3} $\pm 1.82\%$
$n = 2$	4.27×10^{-3} $\pm 9.97\%$	4.36×10^{-3} $\pm 0.87\%$	8.12×10^{-4} $\pm 10.90\%$	8.18×10^{-4} $\pm 1.78\%$
$n = 4$	2.82×10^{-3} $\pm 12.06\%$	2.99×10^{-3} $\pm 0.89\%$	4.96×10^{-4} $\pm 12.88\%$	5.21×10^{-4} $\pm 1.73\%$
$n = 8$	1.18×10^{-3} $\pm 17.91\%$	1.31×10^{-3} $\pm 1.08\%$	1.88×10^{-4} $\pm 18.37\%$	2.10×10^{-4} $\pm 1.77\%$
$n = 16$	—	2.23×10^{-4} $\pm 1.49\%$	—	3.41×10^{-5} $\pm 2.01\%$
$n = 32$	—	5.70×10^{-6} $\pm 2.20\%$	—	8.62×10^{-7} $\pm 2.57\%$
$n = 64$	—	2.76×10^{-9} $\pm 3.51\%$	—	4.18×10^{-10} $\pm 3.75\%$

Table 4 Estimates of γ_n and \mathcal{F}_n in a $TPBS/D/1/k$ Queue

	γ_n		\mathcal{F}_n	
	Std. Sim.	Fast Sim.	Std. Sim.	Fast Sim.
$\alpha = 0.05$	3.07×10^{-2} $\pm 4.04\%$	3.12×10^{-2} $\pm 1.50\%$	2.20×10^{-3} $\pm 5.22\%$	2.22×10^{-3} $\pm 2.83\%$
$\alpha = 0.10$	1.55×10^{-3} $\pm 14.46\%$	1.53×10^{-3} $\pm 1.53\%$	1.61×10^{-4} $\pm 17.54\%$	1.53×10^{-4} $\pm 2.80\%$
$\alpha = 0.15$	6.72×10^{-5} $\pm 58.73\%$	6.39×10^{-5} $\pm 1.55\%$	9.48×10^{-6} $\pm 71.29\%$	7.77×10^{-6} $\pm 2.73\%$
$\alpha = 0.20$	—	2.18×10^{-6} $\pm 1.60\%$	—	3.13×10^{-7} $\pm 2.70\%$
$\alpha = 0.25$		6.09×10^{-8} $\pm 1.65\%$		9.80×10^{-9} $\pm 3.18\%$
$\alpha = 0.30$		1.34×10^{-9} $\pm 1.72\%$		2.44×10^{-10} $\pm 3.12\%$
$\alpha = 0.35$	—	2.26×10^{-11} $\pm 1.71\%$	—	4.58×10^{-12} $\pm 2.54\%$
$\alpha = 0.40$	—	2.86×10^{-13} $\pm 1.79\%$	—	6.42×10^{-14} $\pm 3.03\%$

AUTHOR BIOGRAPHIES

VICTOR F. NICOLA holds the Ph.D. degree in computer science from Duke University, North Carolina. From 1979 he held scientific and research staff positions at Eindhoven University and Duke University. In 1987, he joined IBM T.J. Watson Research Center as a Research Staff Member. Since 1993 he holds an Associate Professor position with the group of Tele-Informatics and Open Systems at the University of Twente. His research interests include performance and reliability modeling of computer and communication systems, queueing theory, fault-tolerance and simulation.

GERTJAN A. HAGESTEIJN holds the M.Sc. degree in computer science from the University of Twente, The Netherlands. He is currently involved in the development of fast simulation techniques for the evaluation of ATM-based telecommunication systems.

On the accelerated simulation of VBR virtual channel multiplexing in a single-server FIFO buffer

M. J. Tunncliffe, D. J. Parish,
Department of Electronic and Electrical Engineering,
Loughborough University, Ashby Road, Loughborough,
Leicestershire, United Kingdom, LE11 3TU.
Tel.: (01509) 228117, Fax.: (01509) 222854.
e-mail: M.J.Tunncliffe2@Lboro.ac.uk
WWW: <http://info.Lboro.ac.uk/departments/el/research/hsn/index.html>

Abstract

While direct *cell-level* simulation accurately predicts congestion in cell-switched networks, excessive run-times are often required to obtain significant results. Methods of *Accelerated* simulation have therefore been developed, examples of which include the *cell rate* technique (which represents the discrete cell-streams as continuous fluids) and the *histogram* method (which merges the multiplexed streams into an aggregate cell-rate histogram and performs independent statistical analysis on each bin). The current work applies both these techniques to a simple ATM multiplexer and explores their respective advantages and drawbacks. While the cell-rate method provides accurate predictions under a rapidly varying bit rate, the histogram method is more successful under quasi-static conditions. This suggests the possibility of a hybrid cell-rate/histogram model which is accurate at both extremes.

Keywords

ATM networks, simulation techniques, statistical analysis.

1 INTRODUCTION

The recent proliferation of cell-switched communication networks has led to increasingly complex problems in their design, evaluation and management. Such problems, many of which arise from congestion as virtual channels are multiplexed, lead to cell-losses and transmission-time *jitter*, the latter being particularly harmful to real-time services such as video.

Various computer-aided techniques have been devised for the analysis of networking problems (see Kurose and Mouftah 1988 and Frost et al. 1988). The most direct approach is *cell-level simulation*, in which network components are directly represented within the software, and cell arrivals and transmissions are mimicked by pseudorandom sequences. Such simulations are highly processor-intensive, and enormous run-times are often required to simulate relatively short periods of operation. For example, the Orwell simulator recently developed at Loughborough University (Parish et al., 1994) can take several hours to simulate one minute of real-time, and since *acceptable* loss rates are of the order of 10^{-9} (i.e. 1 lost cell in 10^9), several weeks may be required to obtain statistically significant characterization.

For this reason, numerous workers have investigated *accelerated* simulation techniques, which allow run-times to be reduced without major loss of accuracy. One example is *variance reduction* which manipulates the statistical properties of a cell-level model in order to reduce the stochastic variability of its output, thus shortening the run-time needed to obtain statistical significance (Frost et al, 1988). However, the current paper concentrates on the following recently-published techniques:

- The *cell rate* method, developed by Pitts et al. (1994 a,b) at Queen Mary & Westfield College, London, represents the various discrete cell-streams applied to the input buffer of an ATM multiplexer as continuous fluids, whose flow-rates are modulated by the bulk-traffic characteristics. This method has been shown to produce accurate cell-loss predictions in the *burst-scale* where the aggregate cell-rate exceeds the channel capacity.
- The *histogram* method, introduced by Skelly et al. (1993) at the University of Columbia, NY, converts the incoming cell-streams of an ATM multiplexer into arrival-rate histograms and convolves them together to form an aggregate histogram. Statistical queueing analysis is applied separately to each histogram bin, and the results are then combined as a weighted sum. It is a fundamental assumption of this model that the system is in statistical quasi-equilibrium, and it is therefore unsuitable for rapidly varying bit-rates.

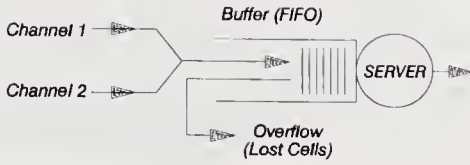
The current paper applies variants both these techniques to a simple two-channel multiplexer. The predictions are compared with the results of a stochastic cell-level simulator and their respective accuracies and run-times are contrasted.

2 CELL-LEVEL SIMULATION

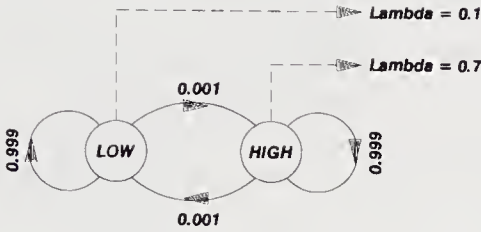
Before the accuracies of any accelerated simulation techniques could be tested, it was first necessary to establish a cell-level simulator against which their predictions could be compared. Figure 1(a) shows a schematic diagram of the ATM multiplexer modelled in the software (which was written in Turbo-C and ran upon a 486-based desktop microcomputer). Time was quantised into *cycles*, during each of which up to one cell could arrive on each input channel and up to one cell could be read by the server. The latter operated in a *geometric* mode, in which there was a constant probability (μ) per cycle of a cell being read.

Each of the two buffer inputs could be fed with any user-defined data-stream. If both channels generated a cell within the same cycle (i.e. a *batch* arrival) both were simultaneously loaded onto the buffer in a randomly selected order (i.e. each cell had equal probability of getting first place in the buffer).

(a) Schematic Diagram of ATM Multiplexer



(b) MMB Source Model for Channel 2



(b) Simulated Cell-Delay Distributions

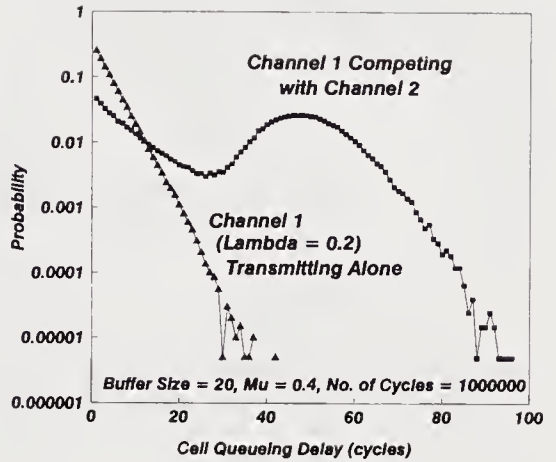


Figure 1: Cell-level simulation of channel interaction in an ATM multiplexer.

Figure 1(c) shows some typical results obtained from the simulator using a buffer-length (N) of 20 and a geometric service-probability of 0.4 per time-slot. Firstly, an unmodulated Bernoulli stream (arrival probability $\lambda=0.2$) was sent through the buffer on Channel 1 with Channel 2 inactive, and the cell-delay distribution was recorded. The experiment was then repeated with an additional 2-state Markov-modulated Bernoulli stream applied to Channel 2 (Figure 1(b)), and the subsequent deterioration of transmission quality (i.e. increased cell-delay) is clearly visible in the results (Figure 1(c)). The upper mode in the cell-delay distribution clearly represents the *burst* component, where the aggregate cell arrival rate exceeds the server capacity and the buffer becomes normally full.

3 HISTOGRAM SIMULATION

The analysis presented in this section assumes that the buffer is in statistical equilibrium, and hence that the equilibrium probabilities $\Pi_0 \dots \Pi_N$ remain constant with time. (Π_n is the probability that the buffer contains n cells).

Statistical Queueing Analysis

If only a single Bernoulli stream is applied then the buffer can be modelled as a discrete-time Geo/Geo/1/N queue, the solution of which is a matter of simple textbook theory. For $0 \leq n < N$, the equilibrium probabilities are given by:

$$\Pi_n = \frac{(1 - \gamma) \cdot \gamma^n}{1 - \frac{\lambda}{\mu} \cdot \gamma^N} \quad \text{where} \quad \gamma = \frac{\lambda (1 - \mu)}{\mu (1 - \lambda)} \quad (1)$$

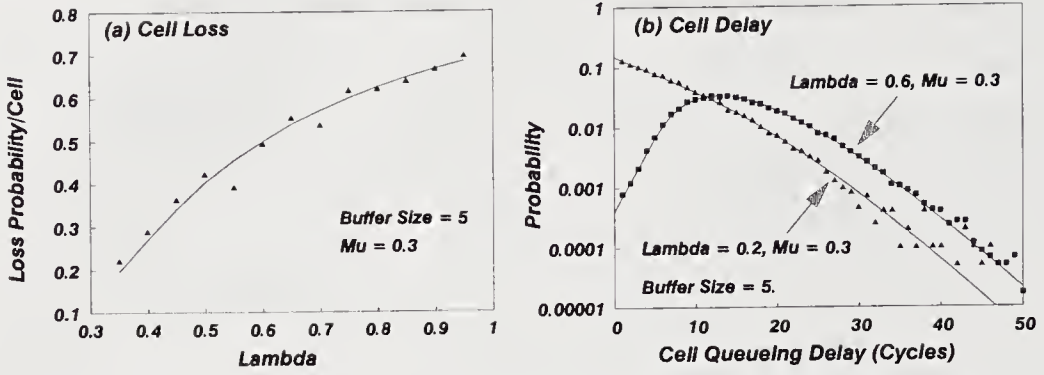


Figure 2: Simulated behaviour of Geo/Geo/1/N buffer compared with analytical model predictions. (Discrete points indicate simulations, solid lines indicate the analytical model.)

while for $n = N$:

$$\Pi_N = \frac{(1 - \gamma)(1 - \lambda) \cdot \gamma^N}{1 - \frac{\lambda}{\mu} \cdot \gamma^N} \quad (2)$$

If an arriving cell finds n ($< N$) cells ahead of it, then it remains in the buffer until the latter has been read $(n+1)$ times. Hence the probability that queueing delay is equal to k cycles is given by

$$P(k) = \sum_{n=0}^{N-1} \Pi_n \mu^{n+1} (1 - \mu)^{k-(n+1)} \binom{k-1}{n} \quad (3)$$

Substituting Eqn.(1) for Π_n and simplifying yields

$$P(k) = \Pi_0 \mu (1 - \mu)^{k-1} \left[(1 - \lambda)^{1-k} - \sum_{i=N}^{k-1} \left[\frac{\lambda}{1 - \lambda} \right]^i \binom{k-1}{i} \right] \quad (4)$$

Since the final term in (assumed zero for $k < N+1$) is a truncated binominal series, it may be replaced by the *incomplete beta function* $I_\lambda(N, k - N)$. The expression may now be re-written:

$$P(k) = \Pi_0 \mu \left[\frac{1 - \mu}{1 - \lambda} \right]^{k-1} \cdot [1 - I_\lambda(N, k - N)] \quad (5)$$

If an incoming cell finds N cells already in the buffer then the latter is full and the new cell must therefore be lost. Hence the loss probability is equal to Π_N and may therefore be computed using Equation 2. Figure 2 compares the analytical cell-loss and cell-delay

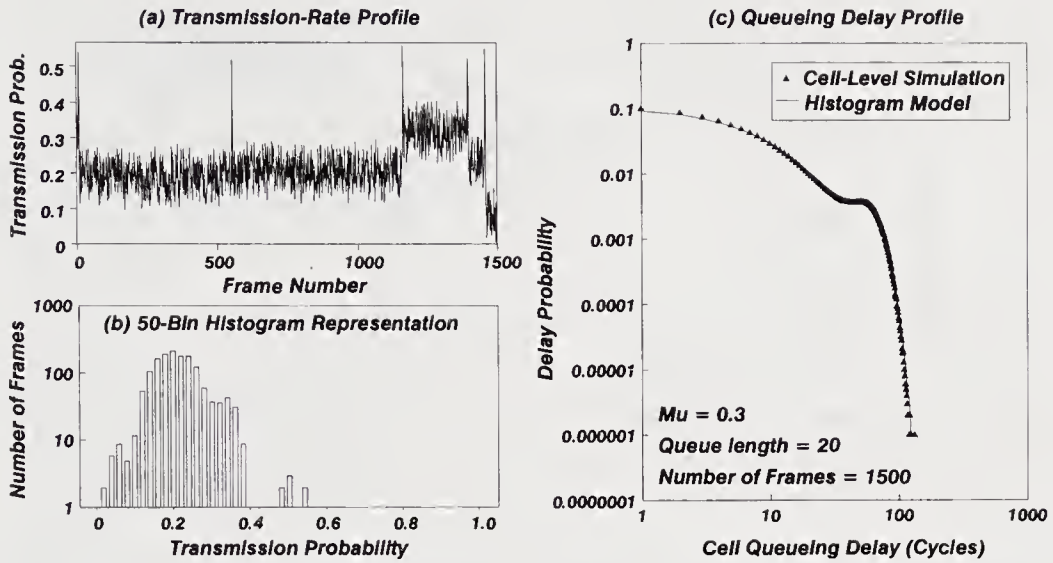


Figure 3: Example of single-channel histogram simulation.

characteristics with the results of cell-level simulations. (Numerical values for the incomplete beta function I_λ were computed using the algorithm supplied in *Numerical Recipes in C* by Press et al. (1988).)

Single Channel Histogram Simulation

Figure 3(a) shows an example of the simulated variable bit rate (VBR) video profiles used in this study. (The video simulation was based upon the output of an experimental VBR codec during the compression of "head-and-shoulders" image sequences. The occasional high cell-rate excursions correspond to scene-changes within the sequence, while the smaller variations indicate activity within individual scenes.) The duration of each video frame was 28 276 cycles, over which the arrival probability λ remained constant. (This period was sufficiently long for the assumption of statistical equilibrium to be approximately valid).

Figure 3(b) shows the same video profile expressed as a 50-bin arrival-probability histogram. Independent statistical queueing analysis was performed upon each bin (for a buffer size of $N=20$ and service rate $\lambda=0.3$), after which the results were weighted according to their relative frequencies, and finally summed to obtain the overall delay and loss characteristics. Figure 3(c) shows the resulting queue-delay distribution (solid lines) compared with a cell-level simulation of the same scenario (discrete points). The cell-loss ratio was computed as 2.56% by the histogram model, compared to 2.14% predicted by the cell-level simulation.

Since the distributions in Figure 3(c) are practically indistinguishable, the histogram curve clearly provides a highly accurate approximation of the simulated data. As the histogram results were obtained in approximately 1% of the cell-level run-time (i.e. 111 seconds compared to 10 494 for the cell-level simulation), the experiment illustrates the potential value of the histogram method as a means of reducing simulation run-time.

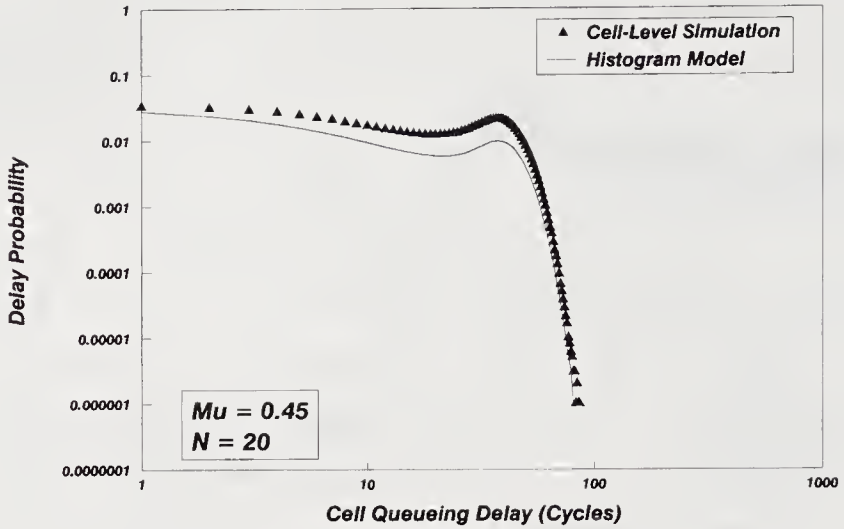


Figure 4: Typical results of two-channel histogram simulation.

Two-Channel Histogram Simulation

Since the primary focus of this paper is the interaction of competing virtual-channels in a common buffer, the above theory must be extended to cover the effects of multiple Bernoulli inputs. If the cell arrival probabilities for channels 1 and 2 are λ_1 and λ_2 respectively and p_n is the probability of n arrivals per cycle, then:

$$p_0 = (1 - \lambda_1)(1 - \lambda_2) = 1 - (\lambda_1 + \lambda_2) + \lambda_1 \lambda_2 \quad (6)$$

$$p_1 = \lambda_1(1 - \lambda_2) + \lambda_2(1 - \lambda_1) = \lambda_1 + \lambda_2 - 2 \cdot \lambda_1 \lambda_2 \quad (7)$$

$$p_2 = \lambda_1 \lambda_2 \quad (8)$$

The resultant arrival stream is an example of a *batch* Bernoulli process (maximum batch size = 2) whose effects upon discrete-time queueing have been analytically studied by Dafermos et al. (1971) and more recently by Hashida et al. (1991). However, the current analysis employs the following simplifying assumption: If λ_1 and λ_2 are both $\ll 1$ then $\lambda_1 \lambda_2$ becomes negligible and the aggregate stream approximates to a standard Bernoulli process with arrival probability $(\lambda_1 + \lambda_2)$.

Figure 4 shows some typical histogram and cell-level results for the interaction of two independent VBR streams in a common buffer. The latter were initially converted into individual arrival-rate histograms, which were then convolved together to form the aggregate histogram. Although a wide divergence exists in some parts of the graph, the general agreement in the shapes of the curves illustrates the value of the technique. In view of the accuracy of the one-channel case, these errors are almost certainly associated with the

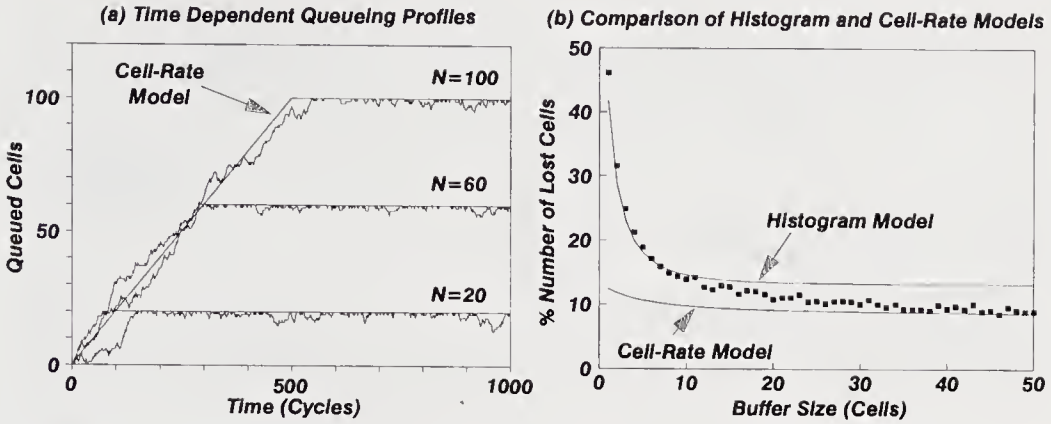


Figure 5: Illustration of cell-rate model, and comparison with the predictions of cell-level and histogram simulations.

aggregation approximation (Section 3) and/or the histogram convolution technique which is, strictly speaking, applicable only to stationary stochastic processes. Extension of the time-domain would therefore be expected to improve prediction accuracy.

4 CELL-RATE METHOD

The cell-rate simulation technique described below is a simplified version of that published by Pitts et al. (1994 a,b). Its main function is to show how the primary properties of this algorithm differ from those of the histogram method described above.

Basically, the cell-rate model ignores the discrete nature of the cell streams, and represents them as continuous fluids modulated by a *burst traffic* profile. The latter is composed of constant cell-rate *bursts*, punctuated by discontinuous cell-rate changes known as *events*. Within each burst, when the buffer is neither full nor empty, the number $n(t)$ of cells in the queue varies with time t (cycles) according to the equation

$$n(t) = n_0 + \left[\sum_{r=1}^k \lambda_r - \mu \right] (t - t_0); \quad 0 < n(t) < N. \quad (9)$$

where t_0 is the time at which the burst began, n_0 is the number of queued cells when $t=t_0$ and k is the number of multiplexed streams. This transient phase ends when the queue becomes full or empty, and n remains equal to N or 0 until the end of the burst. Figure 5(a) shows these *transient* and *steady state* phases compared with the corresponding cell-level simulations for an initially empty Geo/Geo/1/ N queue. When the buffer is full and the aggregate cell-rate exceeds the server capacity, losses occur at a rate of $(\sum \lambda - \mu)$ cells per cycle, and are distributed between Channels 1 and 2 according to the ratio $\lambda_1:\lambda_2$.

Unlike the Pitts et al. model (which handles network events concurrently) the cell-rate simulator program tracks buffer occupancy from burst-to-burst throughout the range of the

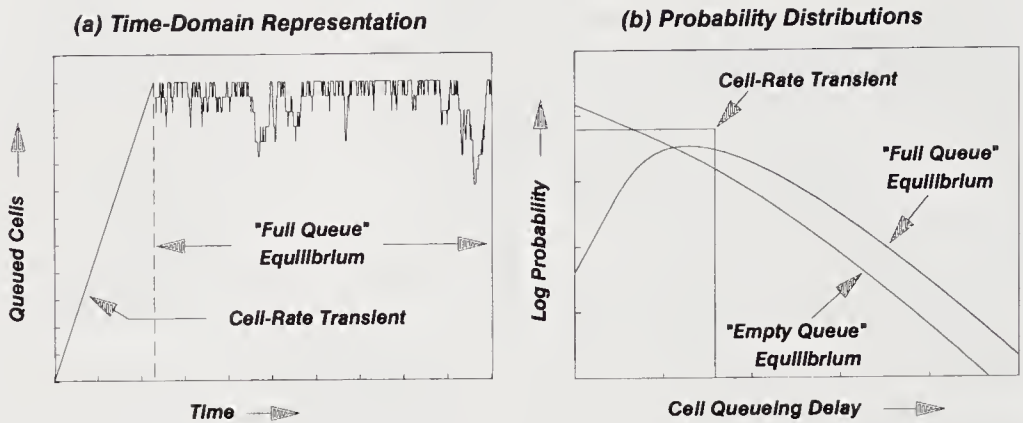


Figure 6: Illustration of proposed hybrid cell-rate/histogram model.

simulation, recording the number of lost cells. Figure 5(b) shows some typical cell-loss characteristics compared with the results of the cell-level and histogram models. During a high arrival-rate burst when $\Sigma\lambda/\mu > 1$ and cell-losses become significant, the length of the expanding queue is constrained by the buffer capacity and the transient phase can be expected to be of the same order as the buffer-filling time, i.e. $t_{\text{tran}} \sim N / (\Sigma\lambda - \mu)$. Hence the observed increase in accuracy with increasing N . However, when N is small and $t_{\text{burst}} \gg t_{\text{tran}}$, transient phenomena can be entirely neglected and the queue assumed to be in equilibrium throughout the simulation. Hence when N becomes small, the histogram model provides the best predictions.

5 CONCLUSIONS AND FUTURE WORK

This paper has presented some early results from an ongoing study of computer-aided communication-network modelling. The initial stage of the work involved the design and testing of a cell-level simulator for a single-server FIFO buffer with a geometric read-time distribution, fed by two independent cell-streams. (This system could provide a module in a full network simulator). The same system was also modelled using the *cell-rate* technique and a *histogram* model based upon statistical queueing theory. Although both these models produced results within significantly shorter run-times than the cell-level simulator, they were found to be accurate only within certain regions of parameter-space. We now consider the possibility of a *hybrid* model, combining the respective virtues of these two algorithms.

Such an algorithm would be required to model the stochastic nature of both the transient and steady-state conditions of operation. Although several transient models are available for the unbounded, continuous-time M/M/1/ ∞ queue (a computationally efficient formula has recently been developed at Bradford University (Bunday, 1995)), the finite capacity of the ATM buffer presents severe theoretical difficulties. One possible solution is illustrated in Figure 6(a): The cell-rate model is applied during all transient phases of operation, while the statistical equilibrium model is used during periods of statistical equilibrium (i.e. when the

value of n predicted by the cell-rate model is either 0 or N). The cell delay distribution in a transient phases might be represented to some degree of accuracy by a rectangular function of width μN and height $1/\mu N$ (Figure 6(b)).

It should be noted that the results represent only the most preliminary findings of an ongoing study of network modelling, and are not intended to form a definitive treatment of the subject. Investigations have so far been confined to a single network component, consisting of a single queue and a single server, under relatively simple traffic-loading conditions (although some of the bulk statistics *were* based upon a realistic VBR video-source model). The model must ultimately be extended to cover a network of many such interconnected units under more generalized traffic, which may include such complicating effects as correlation in the cell-generation process (Skelly, 1994). The validity of the resulting model must then be checked by comparing its predictions against the operational statistics of an actual hardware network under realistic traffic-loading conditions.

6 ACKNOWLEDGEMENTS

The authors wish to thank all members of the High Speed Networks research group at Loughborough University for their suggestions and technical assistance. The work was funded by an EPSRC-ROPA grant.

7 REFERENCES

- Bunday,B. (1995), Dept. of Mathematics, University of Bradford, private communication.
- Dafermos,S.C., Neuts,M.F. (1971) A Single Server Queue in Discrete Time, *Cahiers du Centre D'étude de Rechherche Opérationnelle*, **19**, 23-40.
- Frost,V.S., Wood Larue,W, Shanmugan,K.S. (1988) Efficient Techniques for the Simulation of Computer Communications Networks, *IEEE J-SAC*, **6**, 146-57.
- Hashida,O, Takahashi,Y, Shimogawa,S (1991) Switched Batch Bernoulli Process (SBBP) and the Discrete-Time SBBP/G/1 Queue with Application to Statistical Multiplexer Performance, *IEEE J-SAC*, **9**, 394-401.
- Kurose,J.F., Mouftah,H.T. (1988) Computer-Aided Modelling, Analysis, and Design of Communication Networks, *IEEE J-SAC*, **6**, 130-45.
- Parish,D.J., Rogers,C, Nche,C, Ruiz,I (1994) Modelling the Orwell Network Access Protocol on a Slotted Ring, in *Computer and Telecommunication Systems Performance Engineering* (eds. M.E.Woodward, S.Datta, S.Szumko), Pentech Press, London, 108-13.
- Pitts,J.M., Cuthbert,L.G., Bocci,M., Scharf,E.M. (1994a) Cell Rate Modelling: An Accelerated Simulation Technique for ATM Networks, *ibid.*, 94-107.
- Pitts,J.M., Cuthbert,L.G., Bocci,M, Scharf,E.M. (1994b) An Accelerated Simulation Technique for Modelling Burst-Scale Queueing Behaviour in ATM, *Teletraffic Congress, 14th. International Conference*, **1**, 777-86.
- Press,W.H, Teukolsky,S.A., Vetterling,W.T., Flannery,B.P. (1988) Numerical Recipes in C, Cambridge University Press.
- Skelly,P, Schwartz,M, Dixit,S (1993) A Histogram-Based Model for Video Traffic Behaviour in an ATM Multiplexer, *IEEE/ACM Trans. Networking.*, **1**(4), 446-59.

8 BIOGRAPHIES

Martin J. Tunnicliffe holds B.Eng. and Ph.D. degrees from the Universities of Bradford and Loughborough respectively. His early research was in the field of semiconductor reliability, but he has more recently been involved in the monitoring and analysis of communication networks. He is currently employed as a contract researcher in the High Speed Networks Group at Loughborough University.

David J. Parish holds B.Sc. and Ph.D. degrees from the University of Liverpool. He has worked as a Scientific Officer at the UKAEA Culham Laboratory and as a Demonstrator in the Electrical Engineering Department at Liverpool University. From 1983 he has held the position of Lecturer and later Senior Lecturer in the Department of Electronic and Electrical Engineering at Loughborough University. His research interests concern the management, operation, monitoring and application of High Performance Networks. Specifically, he leads Loughborough's input to the BT funded research programme into the management of high speed networks using SuperJanet.

INDEX OF CONTRIBUTORS

- Ajmone Marsan, M. 175
Arvidsson, Å. 39

Bhabuta, M. 287
Bianco, A. 175
Bose, S.K. 22

Casals, O. 400
Cerdà, L. 400
Chalasani, S. 269
Cigno, R.L. 175
Conti, M. 3

Dagiuklas, A. 358
De Laet, G. 342

Fan, Z. 74
Feng, Y. 233
Fiche, G. 381

García, J. 400
Gelenbe, E. 233
Ghanbari, M. 358
Gravey, A. 57
Gregori, E. 3
Griffiths, J.M. 327

Gustafsson, E. 110

Hagesteijn, G.A. 414
Halberstadt, S. 57
Harrison, P. 287
Hawker, I. 133
Hoeksema, F. 92

Karlsson, G. 110
Kofman, D. 57
Kokkinakis, G. 153
Kouvatsos, D. 287
Kroeze, J. 92

Le Palud, Cl. 381
Lind, C. 39
Liotopoulos, F.K. 269
Logothetis, M. 153

Mang, X. 233
Mars, P. 74
Meyer, J.F. 249
Montagna, S. 249
Munafò, M. 175
Murphy, J. 197
Murphy, L. 197

Naudts, J. 342
Nicola, V.F. 414

Paglino, R. 249
Papanikos, I. 153
Parish, D.J. 431
Pitts, J.M. 327

Rao, T.S. 22
Rouillard, S. 381

Smith, D.G. 133
Srivathsan, K.R. 22

Truffet, L. 215
Tunncliffe, M.J. 431
Tye, B.J. 358

Veitch, P.A. 133

Wilkinson, J. 287
Witters, J. 92

Yin, X.W. 342

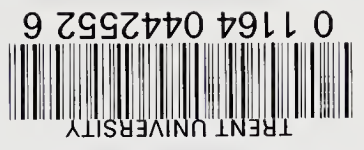
KEYWORD INDEX

- ABR 175
- Accuracy 39
- Asymmetrical Clos networks 269
- Asynchronous Transfer Mode (ATM)
 - switch architectures 287
- ATM 3, 74, 92, 175, 327, 381
 - cell level traffic model 39
 - network performance prediction 233
 - networks 57, 110, 153, 197, 414, 431
 - switch 249, 342
 - switches 269
 - traffic simulation 342
 - virtual paths 133
- B-ISDN 92
- Bandwidth 327
 - allocation 233, 342
- Banyan network 287
- Blocking 381
- Broad-band 381
- Bursty traffic model 39
- Call
 - acceptance 381
 - admission control 233
- CBR 92
- Cell
 - delay variation 381
 - loss 414
- CLOS network 381
- Composite technique 381
- Compound Poisson Process (CPP) 287
- Computer communication networks 269
- Congestion control 197
- Connection admission control 342
- Connectionless services 57
- Diffusion model 233
- Discrete time Markovian models 215
- Dynamic
 - feedback 197
 - routing control 153
- Equivalent capacity 110
- FIR neural networks 74
- GCRA 92
- Generalised Exponential (GE) distribution 287
- Generic cell rate algorithm 342
- Importance sampling 414
- Instantaneous bandwidth available 22
- Lumpability 215
- Markov
 - chain 3
 - decision processes 57
 - modulated
 - Bernoulli Process 39
 - Poisson Process 39
- Maximum Entropy (ME) principle 22, 287
- Measurements 92
- MMBP 39
- MMPP 39
- MPEG 3
- Multi
 - path routing 110
 - stage interconnection network 215
- Multirate networks 269
- Multistage Interconnection Network (MIN) 287
- Narrow-Band 381
- Network parameter control 342
- Nonblocking operation 269
- On-off source 22, 249
- Performance 381
- Policing 327
- Pricing 197
- Quality of Service (QoS) 233, 414
- Queueing, 327
 - Network Model (QNM) 287
 - theory 233
- Rare event simulation 414
- Repetitive-Service (RS) blocking mechanism 287

- Restoration 133
- Routing 133
- Shared buffers 249
- Simulation 92, 175
 - techniques 431
- Statistical
 - analysis 431
 - multiplexer 22
 - multiplexing 3, 381
- Strong ordering 215
- Survivable network design 133
- Switching 327
 - network 381
- TCP 175
- Throughput analysis 92
- Traffic
 - and congestion control 358
 - control 110, 175
 - dispersion 110
 - management 57
 - prediction 74
 - shaping 175
- UPC 92
- Usage parameter control 342
- Variable bit rate video 3
- Veinott's criterion 215
- Virtual path bandwidth control 153

[illegible]

38-297





ATM Networks

Performance Modelling and Evaluation Volume 2

Edited by Demetres Kouvatsos

Unlike many books on Asynchronous Transfer Mode, this text approaches the subject systematically and reflects the state-of-the-art technology being applied throughout the world today. In addition it provides a fundamental source of reference in the ATM field.

The following topics are discussed in detail:

- traffic modelling and characterisation;
- traffic and congestion control;
- bandwidth and admission control;
- ATM switch architecture;
- models of ATM switches;
- routing and optimisation;
- quality of service;
- network management;
- high speed LANs and MANs;
- performance modelling studies

The book presents expanded research papers selected from the Third IFIP Workshop on Performance Modelling and Evaluation of ATM Networks, sponsored by the International Federation for Information Processing (IFIP), and held July 1995, Ilkley, UK. It is ideal for personnel in computer/communication industries, and academic and research staff in computer science and electrical engineering.

Demetres Kouvatsos is a Reader in Computer Systems Modelling at the University of Bradford, UK.

Also available from Chapman & Hall

Performance Modelling and Evaluation of ATM Networks Volume 1

Edited by Demetres D. Kouvatsos

Hardback (0 412 71140 0), 640 pages

Information Network and Data Communication

Edited by Finn A. Aagesen, Harald Botnevik and Dipak Khakhar

Hardback (0 412 75750 8), 461 pages

Intelligent Networks and New Technologies

Edited by Jorgan Norgaard and Villy B. Iversen

Hardback (0 412 78900 0), 308 pages

CHAPMAN & HALL

London • Weinheim • New York • Tokyo • Melbourne • Madras

ISBN 0-412-79200-1



9 780412 792007 >